

# الفهرس

(1) بيئة التصميم Design environment

(2) المتغيرات Variables

(3) الجمل الشرطية Conditional Logic

(4) جمل التكرار والدورات الشرطية Loop

(5) الدوال Methods

(6) المصفوفات Arrays

(7) الأحداث Events

(8) التعامل مع الملفات Manipulating Files

(9) تصحيح أخطاء التطبيق Debugging

(10) التاريخ والوقت في لغة C#

(11) برمجة الكائنات الموجهة Object-Oriented (OOP) Programming

(12) برمجة تطبيقات الويندوز

(13) مشاريع للتعامل مع قواعد البيانات

(الإصدار الثاني من الكتاب )

بإذن الله تعالى

# مقدمة

## بيئة التصميم

### بيئة التصميم Design environment (1)

كل البرامج تحتوى على بيئة التشغيل الخاصة بها وهى بمثابة الحاوية التى تضم كل مكونات البرنامج التى قد يحتاجها المبرمج فى عمله .

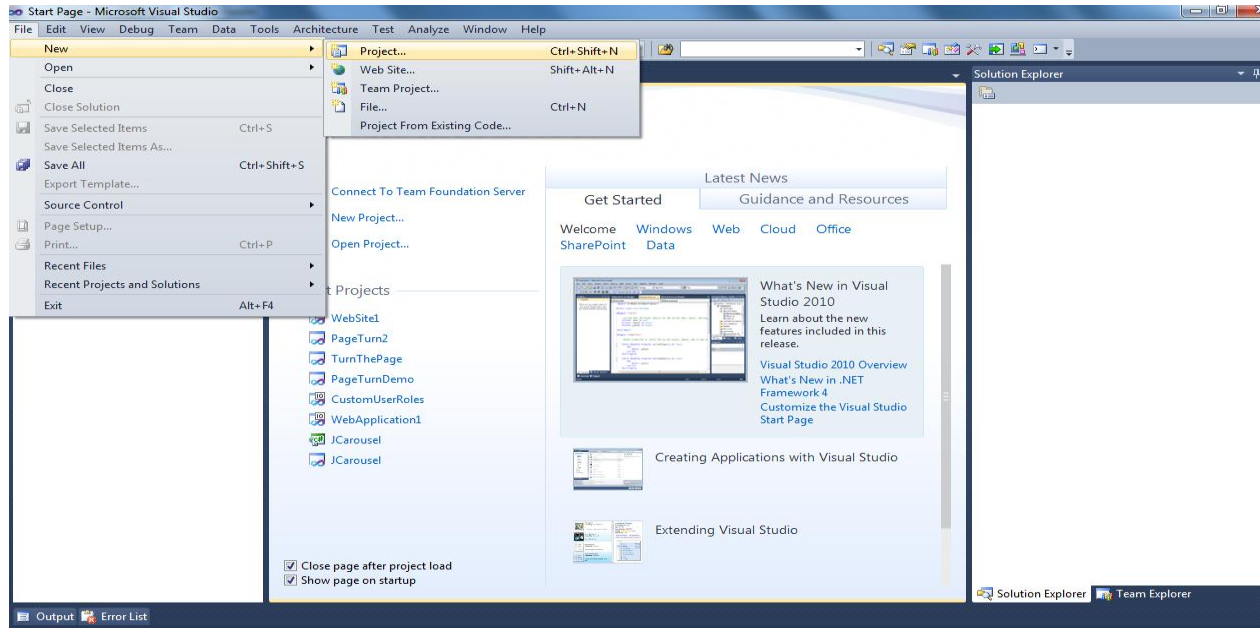
للدخول إلى نافذة التصميم الخاص بمحرر التطوير (Visual Studio)

أولا قم بإنشاء مشروع جديد وذلك عن طريق فتح برنامج

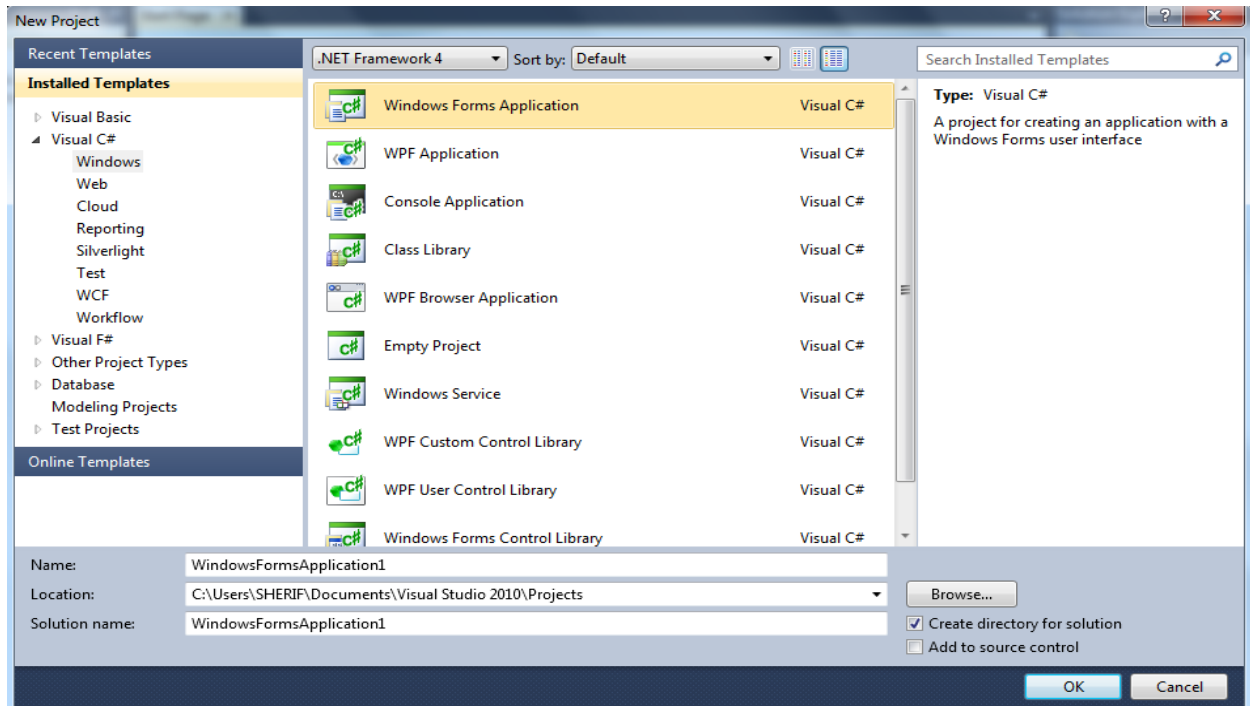
(Visual Studio 2010) أو (Visual Studio 2012) من قائمة Start >-- ثم

الذهاب إلى قائمة File واختيار New Project

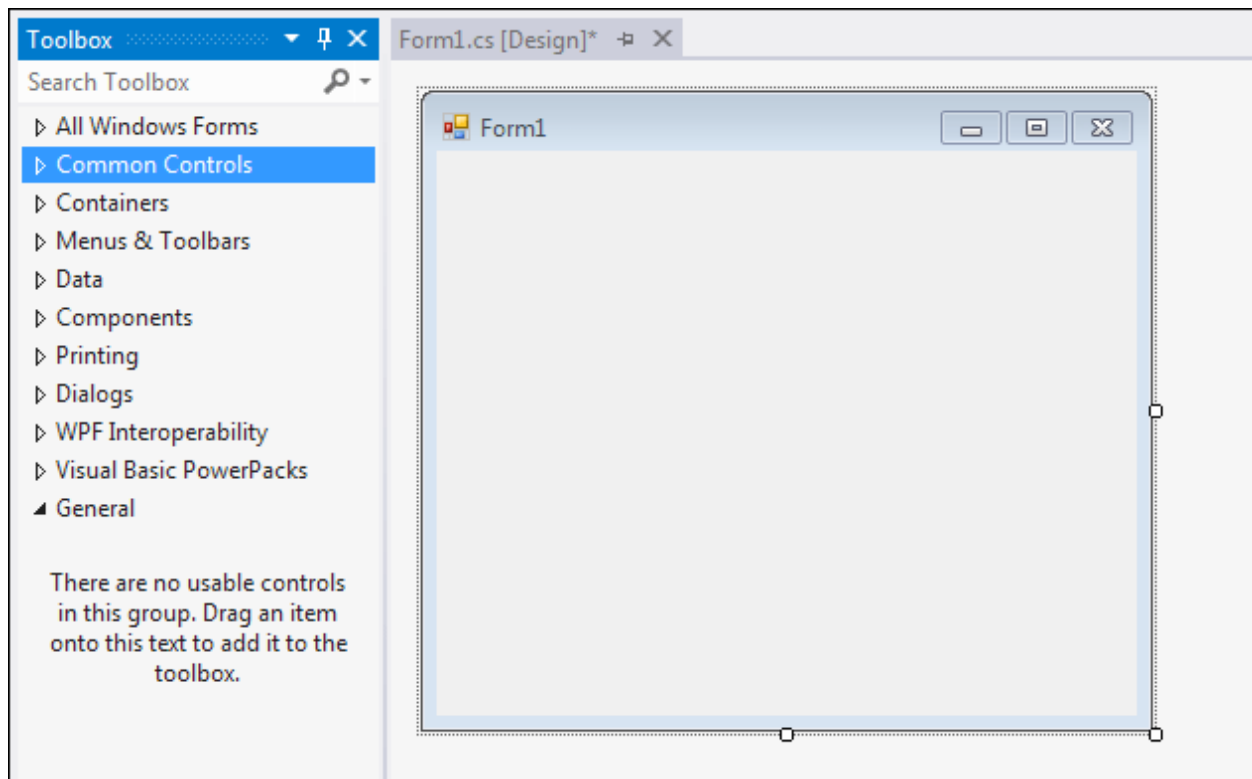
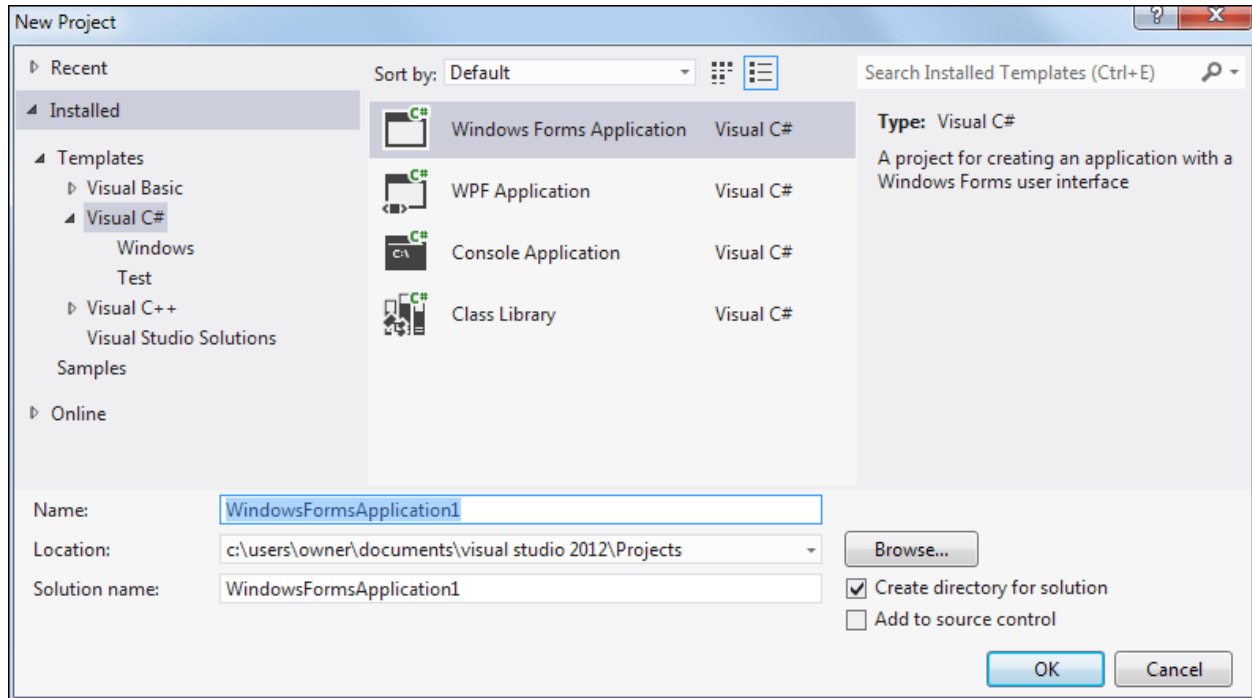
# (Visual Studio 2010)



ستظهر لك هذه النافذة كما يمكنك تغيير اسم البرنامج ومسار حفظه ثم نضغط OK.



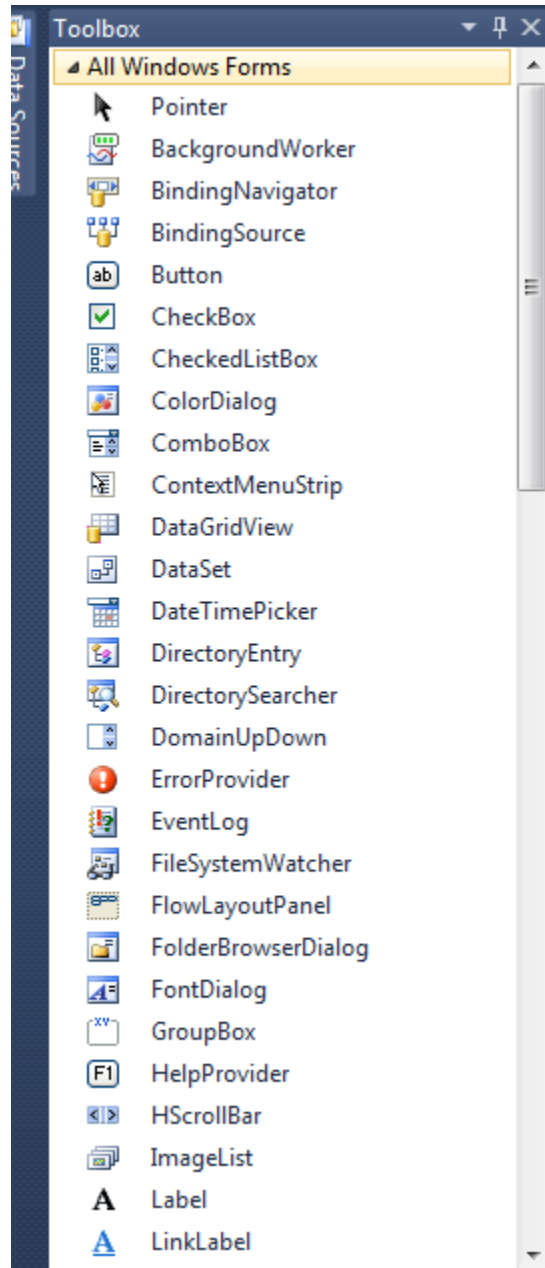
# (Visual Studio 2012)



[illegible]

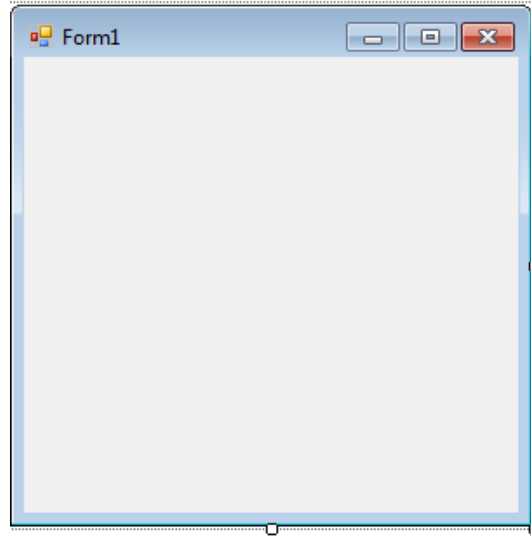
النافذة رقم (1) :

تسمى Toolbox وهى تضم كل الأدوات التى قد يحتاجها برنامجك  
( Buttons, Menus, .... ) إن لم تكن ظاهرة لديك فإذهب إلى قائمة View  
ثم اخترها لكي تظهر أو اكتب بالضغط على الاختصار **Ctrl + Alt + X** كما يلي:



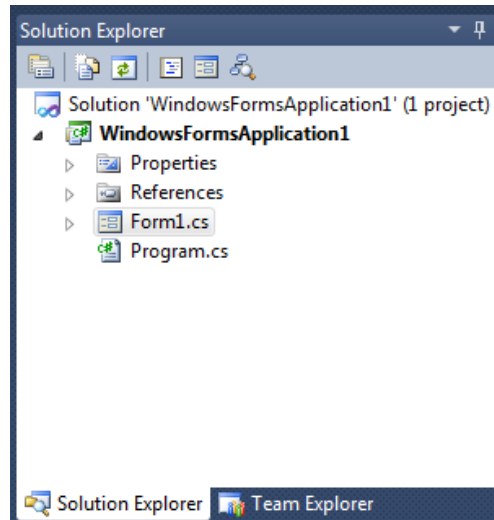
## النافذة رقم (2) :

تسمى **Form** الذى سنضع عليه الأدوات اللازمة لبناء المشروع كما يمكنك إضافة أكثر من واجهة لمشروعك بعد ذلك كما يلي :



## النافذة رقم (3) :

تسمى **Solution Explorer** متصفح المشروع وسمى كذلك لأنه يعرض كل الملفات التى يضمها المشروع ويمكن إظهاره من قائمة **View** واختيار **Solution Explorer** أو الإكتفاء بالضغط على الاختصار **Ctrl + Alt + L** كما يلي :



### النافذة رقم (4) :

تسمى نافذة الخصائص Properties وتحتوى على خصائص الأداة التى نحددها  
ومن خلال هذه النافذة يمكننا تغيير اللون والخلفية والخط وباقى الخصائص , فى  
حالة إخفائها قم بإظهارها من خلال الضغط على مفتاح F4

### النافذة رقم (5) :

**قائمة الأخطاء Error List وتعرض هذه النافذة الأخطاء المرتكبة قبل بدء عملية ترجمة الكود Compilation من خلالها يمكنك معرفة مكان الخطأ ليأتى لك تصحيحه .**

[illegible]



## Variables المتغيرات (2)

### 2.1 المتغيرات النصية Strings

المتغيرات النصية تتكون من مجموعة الحروف والأرقام الموجودة في لوحة المفاتيح بجهاز الكمبيوتر .

لتعريف متغير نصي كالآتي :

**(Type) Variable Name = Value**

**String strVariable = "Internet";**

### برنامج (1)

أنشئ مشروع جديد من قائمة **File -> New -> Project** من فيجوال ستوديو

واختر نوع المشروع **Windows Forms Application** باسم **Strings** .

ضع على الفورم التي تظهر لك الأدوات التالية وأعطى لها الخصائص الآتية:



## 1- Label:

Name: label1

Text: الإسم

Font Name: Times New Roman, 14pt, style=Bold

Size: 14

Bold: True

## 2- Button:

Name: button1

Text: إظهار محتويات مربع النص

Font Name: Times New Roman, 14pt, style=Bold

Size: 14

Bold: True

## 3- TextBox1

Name: textBox1

Text: إظهار محتويات مربع النص

Font Name: Times New Roman, 14pt, style=Bold

Size: 14

Bold: True

Right-to-left: Yes

بالضغط مرتين بالماوس على Button الخاص بإظهار النص ينشئ لنا

الحدث Button1\_Click()

قم بكتابة الكود التالي بداخله كالتالى :

```
private void button1_Click(object sender, EventArgs e)
{
    /*
        This is a Comment
    */

    //This is a String Example

    string firstName;
    firstName = textBox1.Text;

    MessageBox.Show("محتوى مربع النص هو " + firstName);
}
```

لاحظ عملية دمج النصوص فى MessageBox بالعلامة + وهى تعمل

على دمج محتويات النصوص مع المتغيرات النصية كما هو موضح بالكود.

كما تلاحظ كتابة التعليقات على الكود أعلى باللون الأخضر ولعمل التعليقات  
Comments بلغة C# يتم كتابة // فى أول السطر الذى تكتبه هذا لكتابة

سطر واحد من التعليق أما اذا رغبت فى كتابة عدة أسطر من التعليقات لفهم  
مايتم عمله بالكود وإسترجاع ما يتم تنفيذه يتم كتابة

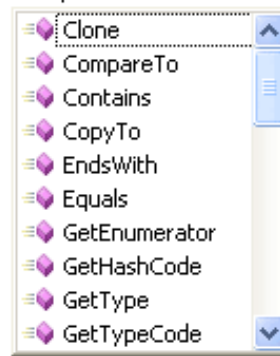
\*/ يتم كتابة أسطر التعليقات هنا \*

## عمليات على المتغيرات النصية (String Manipulation) :

1 - لإجراء عمليات على المتغير النصي بحيث يظهر بالحروف الكبيرة (Capital) تستخدم الدالة `ToUpper()` لذلك .

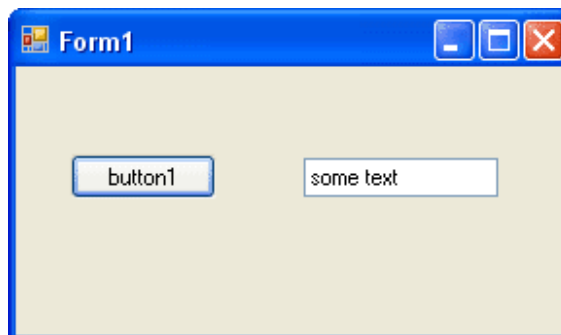
```
private void button1_Click(object sender, EventArgs e)
{
    string stringVar = textBox1.Text;

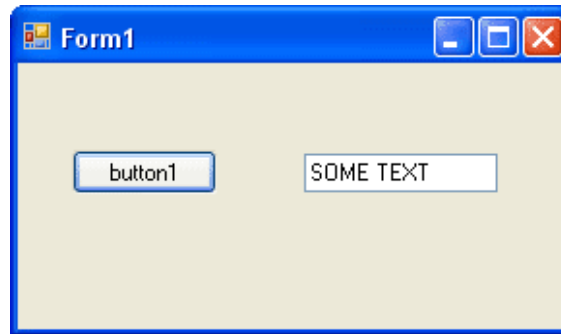
    textBox1.Text = stringVar.
}
```



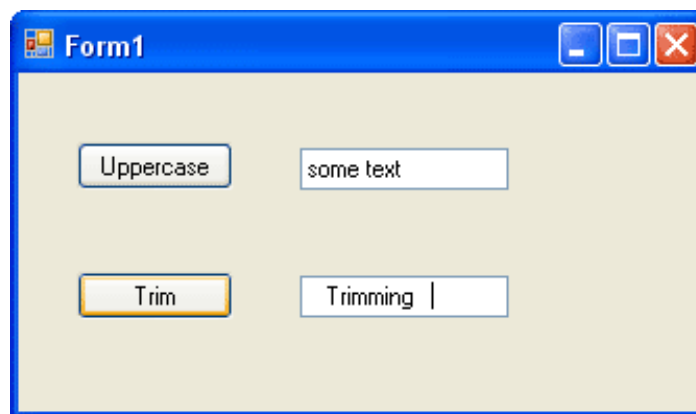
```
private void button1_Click(object sender, EventArgs e)
{
    string stringVar = textBox1.Text;

    textBox1.Text = stringVar.ToUpper();
}
```





2 لحذف المسافات الفارغة في بداية النص وآخره يتم استخدام الدالة Trim()



```
private void button2_Click(object sender, EventArgs e)
{
    string stringTrim = textBox2.Text;

    stringTrim = stringTrim.Trim();

    int stringLength = stringTrim.Length;

    MessageBox.Show( stringLength.ToString() );
}
```

### 3 لحذف المسافات الفارغة أو الحروف فى بداية النص يتم استخدام الدالة TrimStart()

```
stringTrim = stringTrim.TrimStart( trimChars );
```

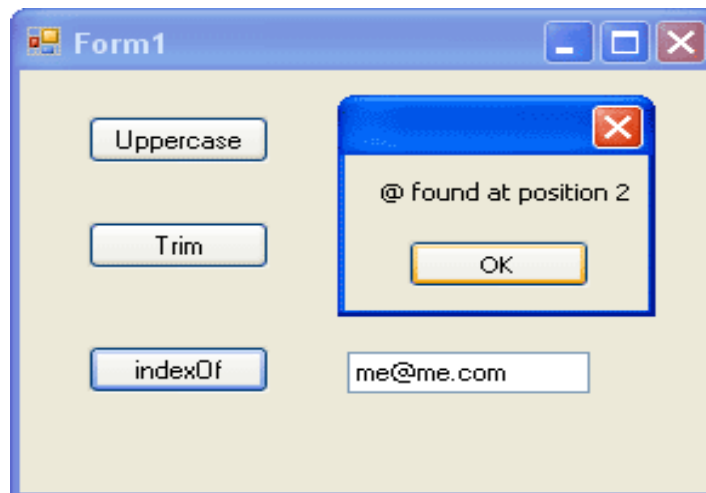
### 4 للبحث عن حرف أو مجموعة حروف فى النص يتم استخدام الدالة Contains()

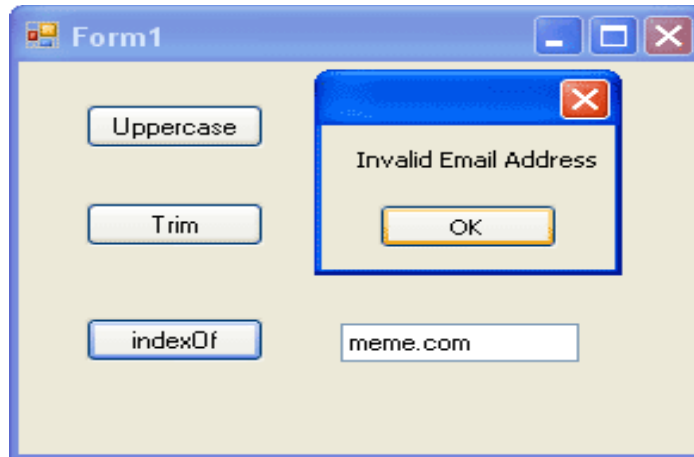
```
private void button4_Click(object sender, EventArgs e)
{
    string stringVar = "Some Text---";

    if (stringVar.Contains("-"))
    {
        MessageBox.Show("TRUE");

        //CALL HYPHEN REMOVE METHOD HERE
    }
}
```

### 5 لمعرفة موقع حرف معين من بين مجموعة حروف يتم استخدام الدالة IndexOf()



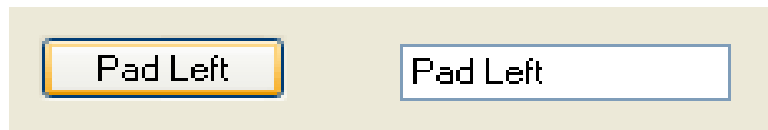


6 لإضافة حرف أو مجموعة من حروف لنص معين يتم استخدام الدالة `Insert()`

```
string someText = "Some Text";
```

```
someText = someText.Insert( 5, "More " );
```

7 لإضافة حرف أو مجموعة من حروف لنص معين يتم استخدام الدالة `PadLeft()`



```
string paddingLeft = textBox5.Text;
```

```
paddingLeft = paddingLeft.PadLeft( 20 );
```



```
paddingLeft = paddingLeft.PadLeft(20 , '*');
```



## 8 - لحذف حرف أو مجموعة من حروف لنص معين يتم استخدام الدالة Remove()

```
string oldString = "some text text text";  
MessageBox.Show(oldString);  
string newString = oldString.Remove(10, 9);  
MessageBox.Show(newString);
```

## 9 - لإحلال حرف أو مجموعة من الحروف محل نص معين تستخدم الدالة Replace(old\_Word , new\_Word)

```
string spellingError = "mistak";  
spellingError = spellingError.Replace(spellingError,  
"mistake");
```

## 10 - للحصول على مقطع من نص معين يبدأ من حرف معين نستخدم الدالة



## Substring(start\_position, no\_of\_chars\_to\_grab)

```
private void button11_Click(object sender, EventArgs e)
{
    string emailAddress = "me@me.com";
    string result = "";

    result = emailAddress.Substring(5, 4);

    if (result == ".com")
    {
        MessageBox.Show("Email Address OK");
    }
    else
    {
        MessageBox.Show("Bad Email Address");
    }
}
```

**11 - دالة تستخدم لفصل مجموعة من الحروف من نص معين بناءً على حرف معين ويمكنك وضع النص الناتج في مصفوفة تسمى الدالة Split()**

```
private void button12_Click(object sender, EventArgs e)
{
    string lineOfText = "item1, item2, item3";

    string[] wordArray = lineOfText.Split(',');

    MessageBox.Show( wordArray[0] );
    MessageBox.Show( wordArray[1] );
    MessageBox.Show( wordArray[2] );
}
```

**12 - دالة تستخدم لتجميع مجموعة من الحروف من نص معين بناءً على حرف معين تسمى الدالة Join()**

```
private void button13_Click(object sender, EventArgs e)
{
    string lineOfText = "item1, item2, item3";
    string[] wordArray = lineOfText.Split(',');

    string joinedText;

    joinedText = String.Join("-", wordArray);

    MessageBox.Show(joinedText);
}
```

## 2.2 المتغيرات الرقمية Integers

هي متغيرات رقمية ولا تقبل نصوص ولا فواصل عشرية  
ويتم تعريفها كالتالي :

**(Type) Variable Name = Value**

**int intVariable = 10;**

### 2.3 المتغيرات الرقمية Float

هي متغيرات رقمية ولا تقبل نصوص وتقبل فواصل عشرية وتتسع لسبعين  
خانات من الأرقام ولا بد من كتابة الحرف F نهاية القيمة المعطاة للمتغير  
وحجمها من ذاكرة الكمبيوتر هو :

**Float:  $1.5 \times 10^{-45}$  to  $3.4 \times 10^{38}$**

ويتم تعريفها كالتالي :

**(Type) Variable Name = Value**

**float floatVariable = 10.56F;**

### 2.4 المتغيرات الرقمية Double

هي متغيرات رقمية ولا تقبل نصوص وتقبل فواصل عشرية وتتسع لـ 16  
خانة من الأرقام

وحجمها من ذاكرة الكمبيوتر هو :

**Double:  $5.0 \times 10^{-324}$  to  $1.7 \times 10^{308}$**

ويتم تعريفها كالتالي :

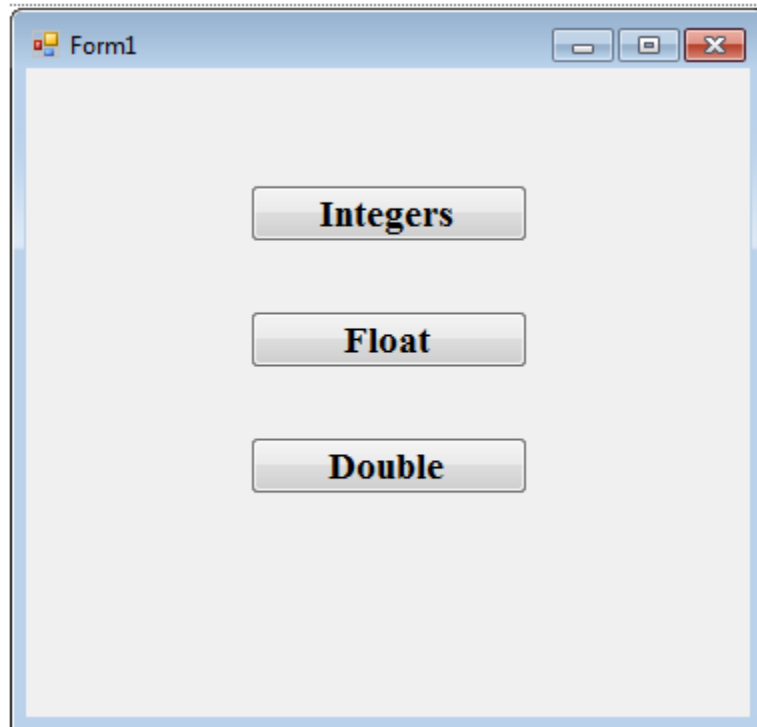
**(Type) Variable Name = Value**

**doubledoubleVariable = 10.56;**

## برنامج (2)

أنشئ مشروع جديد من قائمة **File -> New -> Project** من فيجوال ستوديو  
واختر نوع المشروع **Windows Forms Application** باسم **Numbers** .

ضع على الفورم التي تظهر لك الأدوات التالية وأعطى لها الخصائص الآتية:



## 1- Button:

Name: button1

Text: Integers

Font Name: Times New Roman, 14pt, style=Bold

Size: 14

Bold: True

## 2- Button:

Name: button2

Text: Float

Font Name: Times New Roman, 14pt, style=Bold

Size: 14

Bold: True

### 3- Button:

Name: button3

Text: Double

Font Name: Times New Roman, 14pt, style=Bold

Size: 14

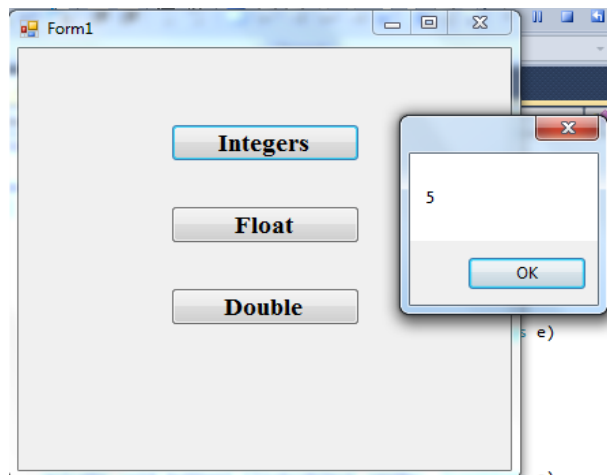
Bold: True

بالضغط مرتين بالماوس على Button Integers ينشئ لنا

الحدث Button1\_Click ()

قم بكتابة الكود التالي بداخله كالتالي :

```
private void button1_Click(object sender, EventArgs e)
{
    int myInteger = 5;
    MessageBox.Show(myInteger.ToString());
}
```

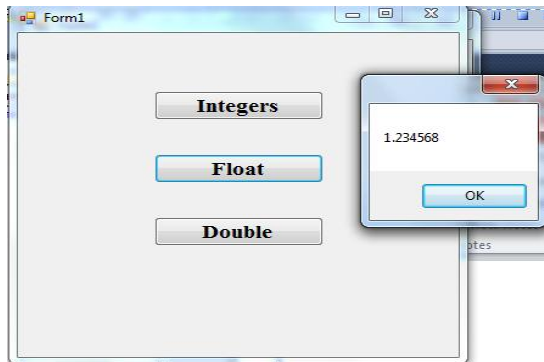


بالضغط مرتين بالماوس على Button Float ينشئ لنا

الحدث Button2\_Click ()

قم بكتابة الكود التالي بداخله كالتالي :

```
private void button2_Click(object sender, EventArgs e)
{
    float myFloat = 1.2345678F;
    MessageBox.Show(myFloat.ToString());
}
```

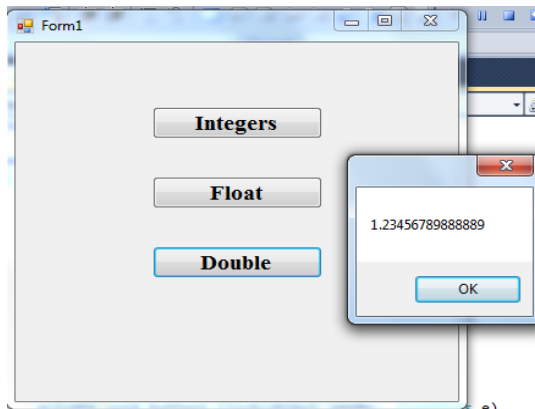


بالضغط مرتين بالماوس على Button Double ينشئ لنا

الحدث Button3\_Click ()

قم بكتابة الكود التالي بداخله كالتالي :

```
private void button3_Click(object sender, EventArgs e)
{
    double myDouble = 1.23456789888888888888;
    MessageBox.Show(myDouble.ToString());
}
```



## Conditional Logic (3) الجمل الشرطية

### 1 جملة IF :

تقوم باختبار الشرط إذا تحقق الشرط و كان صحيحا تنفذ الجمل التي بعدها بداخل الأقواس {} وإذا كان هناك شرط ثانى لجملة IF يسمى Else if إذا تحقق الشرط و كان صحيحا تنفذ الجمل التي بعدها بداخل الأقواس {} أما إذا لم يتحقق نفذ الجملة بعد الأقواس مباشرة لو كان هناك else تنفذ إذا لم يتم تنفيذ أى من الشروط.

**IF (Condition)**

```
{  
    Statements;  
}
```

**IF (Condition)**

```
{  
    Statements;  
}
```

**Else if (Condition)**

```
{  
    Statements;  
}
```



**Else**

**{**

**Statements;**

**}**

## **برنامج (2)**

أنشئ مشروع جديد من قائمة **File -> New -> Project** من فيجوال ستوديو  
واختر نوع المشروع **Windows Forms Application** باسم **Calculator** .

ضع على الفورم التي تظهر لك الأدوات التالية وأعطى لها الخصائص الآتية:

### **1- Button:**

**Name: button1**

**Text: Integers**

**Font Name: Times New Roman, 14pt, style=Bold**

**Size: 14**

**Bold: True**

### **2- Button:**

**Name: button2**

**Text: Integers**

**Font Name: Times New Roman, 14pt, style=Bold**

**Size: 14**

**Bold: True**

### **3- Button:**

**Name: button3**

**Text: Integers**

**Font Name: Times New Roman, 14pt, style=Bold**

**Size: 14**

**Bold: True**

### **4- Button:**

**Name: button4**

**Text: Integers**

**Font Name: Times New Roman, 14pt, style=Bold**

**Size: 14**

**Bold: True**

### **5- Button:**

**Name: button5**

**Text: Integers**

**Font Name: Times New Roman, 14pt, style=Bold**

**Size: 14**

**Bold: True**

### **6- Button:**

**Name: button6**

**Text: Integers**

**Font Name: Times New Roman, 14pt, style=Bold**

**Size: 14**

**Bold: True**

## **7- Button:**

**Name: button7**

**Text: Integers**

**Font Name: Times New Roman, 14pt, style=Bold**

**Size: 14**

**Bold: True**

## **8- Button:**

**Name: button8**

**Text: Integers**

**Font Name: Times New Roman, 14pt, style=Bold**

**Size: 14**

**Bold: True**

## **9- Button:**

**Name: button9**

**Text: Integers**

**Font Name: Times New Roman, 14pt, style=Bold**

**Size: 14**

**Bold: True**

## **10- Button:**

**Name: btnZero**

**Text: Integers**

**Font Name: Times New Roman, 14pt, style=Bold**

**Size: 14**

**Bold: True**

## **11- Button:**

**Name: btnFraction**

**Text: Integers**

**Font Name: Times New Roman, 14pt, style=Bold**

**Size: 14**

**Bold: True**

## **12- Button:**

**Name: btnAdd**

**Text: Integers**

**Font Name: Times New Roman, 14pt, style=Bold**

**Size: 14**

**Bold: True**

## **13- Button:**

**Name: btnSubtract**

**Text: Integers**

**Font Name: Times New Roman, 14pt, style=Bold**

**Size: 14**

**Bold: True**

## **14- Button:**

**Name: btnMultiply**

**Text: Integers**

**Font Name: Times New Roman, 14pt, style=Bold**

**Size: 14**

**Bold: True**

## **15- Button:**

**Name: btnDivide**

**Text: Integers**

**Font Name: Times New Roman, 14pt, style=Bold**

**Size: 14**

**Bold: True**

## **16- Button:**

**Name: btnEqual**

**Text: Integers**

**Font Name: Times New Roman, 14pt, style=Bold**

**Size: 14**

**Bold: True**

## **17- Button:**

**Name: btnClear**

**Text: Integers**

**Font Name: Times New Roman, 14pt, style=Bold**

**Size: 14**

**Bold: True**

## 18- TextBox1

**Name: txtDisplay**

**Font Name: Times New Roman, 14pt, style=Bold**

**Size: 14**

**Bold: True**

**Right-to-left: Yes**

The image shows a screenshot of a Windows application window titled "Form1". Inside the window is a calculator interface. At the top is a text box for displaying the result. Below it is a grid of buttons: a row with buttons for 1, 2, 3, and +; a row with buttons for 4, 5, 6, and -; a row with buttons for 7, 8, 9, and X (multiplication); a row with buttons for 0, . (decimal), and / (division); and a final row with a "Clear" button and an = (equals) button. All buttons have a light gray gradient and a thin black border. The text on the buttons is in a bold, black, serif font.

وبعد الضغط مرتين على كل حدث خاص بالأزرار على الفورم أكتب سطور الكود التالية :

```
privatevoid button1_Click(object sender, EventArgs e)
{
    txtDisplay.Text = txtDisplay.Text + button1.Text;
}

privatevoid button2_Click(object sender, EventArgs e)
{
    txtDisplay.Text = txtDisplay.Text + button2.Text;
}

privatevoid button3_Click(object sender, EventArgs e)
{
    txtDisplay.Text = txtDisplay.Text + button3.Text;
}

privatevoid button4_Click(object sender, EventArgs e)
{
    txtDisplay.Text = txtDisplay.Text + button4.Text;
}

privatevoid button5_Click(object sender, EventArgs e)
{
    txtDisplay.Text = txtDisplay.Text + button5.Text;
}

privatevoid button6_Click(object sender, EventArgs e)
{
    txtDisplay.Text = txtDisplay.Text + button6.Text;
}

privatevoid button7_Click(object sender, EventArgs e)
{
    txtDisplay.Text = txtDisplay.Text + button7.Text;
}

privatevoid button8_Click(object sender, EventArgs e)
{
    txtDisplay.Text = txtDisplay.Text + button8.Text;
}

privatevoid button9_Click(object sender, EventArgs e)
{
    txtDisplay.Text = txtDisplay.Text + button9.Text;
}

privatevoid btnFraction_Click(object sender, EventArgs e)
{
    txtDisplay.Text = txtDisplay.Text + btnFraction.Text;
}

privatevoid btnZero_Click(object sender, EventArgs e)
{

```

```

        txtDisplay.Text = txtDisplay.Text + btnZero.Text;
    }

private void btnClear_Click(object sender, EventArgs e)
{
    txtDisplay.Clear();
}

//Calculator Variables

double total1 = 0;

double total2 = 0;

bool plusButton = false;
bool subtractButton = false;
bool multiplyButton = false;
bool divideButton = false;

private void btnAdd_Click(object sender, EventArgs e)
{
    total1 = double.Parse(txtDisplay.Text) + total1 ;
    txtDisplay.Clear();

    //TRUE
    plusButton = true ;
    subtractButton = false;
    multiplyButton = false;
    divideButton = false;

}

private void btnEqual_Click(object sender, EventArgs e)
{
    if (plusButton == true )
    {
        total2 = total1 + double.Parse(txtDisplay.Text) ;
    }
    elseif (subtractButton ==true)
    {
        total2 = total1- double.Parse(txtDisplay.Text) ;
    }
    elseif (multiplyButton == true)
    {
        total2 = total1 *double.Parse(txtDisplay.Text);
    }
}

```



```

elseif (divideButton == true)
{
    total2 = total1 / double.Parse(txtDisplay.Text);
}

txtDisplay.Text = total2.ToString();
total1 = 0;

}

privatevoid btnSubtract_Click(object sender, EventArgs e)
{
    total1 = double.Parse(txtDisplay.Text)- total1 ;
    txtDisplay.Clear();

    plusButton = false;

//TRUE
    subtractButton = true;
    multiplyButton = false;
    divideButton = false;
}

privatevoid btnMultiply_Click(object sender, EventArgs e)
{
    total1 = double.Parse(txtDisplay.Text);
    txtDisplay.Clear();

    plusButton = false;
    subtractButton = false;

//TRUE
    multiplyButton = true;
    divideButton = false;
}

privatevoid btnDivide_Click(object sender, EventArgs e)
{
    total1 = double.Parse(txtDisplay.Text) ;
    txtDisplay.Clear();

    plusButton = false;
    subtractButton = false;
    multiplyButton = false ;

//TRUE
    divideButton = true;
}

```

## 2 جملة Switch

تقوم باختبار الشرط بناءاً على مجموعة من الحالات تسمى Cases إذا تحققت حالة من الحالات بداخل جملة Switch يتم تنفيذ ما بداخل Case ويتم الخروج من جملة Switch بكلمة **break;**

Switch ()

```
{  
  
    Case "Your_Test_Here_1":  
        // Your Code Here!  
        Break;  
  
    Case "Your_Test_Here_2":  
        // Your Code Here!  
        Break;  
  
    Case "Your_Test_Here_2":  
        // Your Code Here!  
        Break;  
  
    Default:  
        // Your Code Here!  
        Break;  
  
}
```

## برنامج (4)

أنشئ مشروع جديد من قائمة **File -> New -> Project** من فيجوال ستوديو

واختار نوع المشروع **Windows Forms Application** باسم

### **. Calculator Using Switch Statement**

ضع على الفورم التي تظهر لك الأدوات التالية وأعطى لها نفس الخصائص السابقة ثم أضف الكود التالي :

```
privatevoid button1_Click(object sender, EventArgs e)
{
    txtDisplay.Text = txtDisplay.Text + button1.Text;
}

privatevoid button2_Click(object sender, EventArgs e)
{
    txtDisplay.Text = txtDisplay.Text + button2.Text;
}

privatevoid button3_Click(object sender, EventArgs e)
{
    txtDisplay.Text = txtDisplay.Text + button3.Text;
}

privatevoid button4_Click(object sender, EventArgs e)
{
    txtDisplay.Text = txtDisplay.Text + button4.Text;
}

privatevoid button5_Click(object sender, EventArgs e)
{
    txtDisplay.Text = txtDisplay.Text + button5.Text;
}

privatevoid button6_Click(object sender, EventArgs e)
{
    txtDisplay.Text = txtDisplay.Text + button6.Text;
}

privatevoid button7_Click(object sender, EventArgs e)
{
    txtDisplay.Text = txtDisplay.Text + button7.Text;
}

privatevoid button8_Click(object sender, EventArgs e)
{

```

```

        txtDisplay.Text = txtDisplay.Text + button8.Text;
    }

privatevoid button9_Click(object sender, EventArgs e)
{
    txtDisplay.Text = txtDisplay.Text + button9.Text;
}

privatevoid btnFraction_Click(object sender, EventArgs e)
{
    txtDisplay.Text = txtDisplay.Text + btnFraction.Text;
}

privatevoid btnZero_Click(object sender, EventArgs e)
{
    txtDisplay.Text = txtDisplay.Text + btnZero.Text;
}

privatevoid btnClear_Click(object sender, EventArgs e)
{
    txtDisplay.Clear();
}

//Calculator Variables

double total1 = 0;

double total2 = 0;

bool plusButton = false;
bool subtractButton = false;
bool multiplyButton = false;
bool divideButton = false;

string theOperator;

privatevoid btnAdd_Click(object sender, EventArgs e)
{
    total1 = double.Parse(txtDisplay.Text) + total1 ;
    txtDisplay.Clear();

    theOperator = "+";

//TRUE
    plusButton = true ;
    subtractButton = false;
    multiplyButton = false;
    divideButton = false;

}

```

```

privatevoid btnEqual_Click(object sender, EventArgs e)
{

    switch (theOperator)
    {

        case "+":
            total2 = total1 + double.Parse(txtDisplay.Text);
            break;

        case "-":
            total2 = total1 - double.Parse(txtDisplay.Text);
            break;

        case "*":
            total2 = total1 * double.Parse(txtDisplay.Text);
            break;

        case "/":
            total2 = total1 / double.Parse(txtDisplay.Text);
            break;

        default:
            break;

    }

    txtDisplay.Text = total2.ToString();
    total1 = 0;

}

privatevoid btnSubtract_Click(object sender, EventArgs e)
{

    total1 = double.Parse(txtDisplay.Text)- total1 ;
    txtDisplay.Clear();

    theOperator = "-";

    plusButton = false;

//TRUE
    subtractButton = true;
    multiplyButton = false;
    divideButton = false;

}

privatevoid btnMultiply_Click(object sender, EventArgs e)
{

    total1 = double.Parse(txtDisplay.Text);
    txtDisplay.Clear();

    theOperator = "*";

```

```
        plusButton = false;
        subtractButton = false;
//TRUE
        multiplyButton = true;
        divideButton = false;
    }

private void btnDivide_Click(object sender, EventArgs e)
{
    total1 = double.Parse(txtDisplay.Text) ;
    txtDisplay.Clear();

    theOperator = "/";

    plusButton = false;
    subtractButton = false;
    multiplyButton = false ;
//TRUE
    divideButton = true;
}
```

## (4) جمل التكرار و الدورات الشرطية Loops

### 1- جملة For() :

تقوم هذه الجملة بتكرار عدد معين من الجمل بناءا على عدد معين يتم تحديده داخل جملة For

```
for (StartValue ; End_Value ; Update_Expression)
{
    Statements;
}

for (int i = 0; i < length; i++)
{
    Statements;
}
```

### برنامج (5)

أنشئ مشروع جديد من قائمة File -> New -> Project من فيجوال ستوديو  
واختر نوع المشروع Windows Forms Application باسم

.For Loop

ضع على الفورم التي تظهر لك الأدوات التالية وأعطى لها الخصائص الآتية:

#### 1- Button:

Name: button1

Text: For Loop

**Font Name: Times New Roman, 14pt, style=Bold**

**Size: 14**

**Bold: True**

## **2- Textbox:**

**Name: textBox1**

**Font Name: Times New Roman, 14pt, style=Bold**

**Size: 14**

**Bold: True**

**Right-to-left: Yes**

## **3- Textbox:**

**Name: textBox2**

**Right-to-left: Yes**

**Font Name: Times New Roman, 14pt, style=Bold**

**Size: 14**

**Bold: True**

## **4- Label:**

**Name: label1**

**Text : بداية التكرار**

**Font Name: Times New Roman, 14pt, style=Bold**

**Size: 14**

**Bold: True**



#### 4- Label:

Name: label2

Text : نهاية التكرار

Font Name: Times New Roman, 14pt, style=Bold

Size: 14

Bold: True

#### 5- Listbox:

Name: listBox1

Right-to-left: Yes

Font Name: Times New Roman, 14pt, style=Bold

Size: 14

Bold: True

The screenshot shows a Windows application window titled "Form1". Inside the window, there are two text input fields. The first field is labeled "بداية التكرار" (Start of loop) and the second is labeled "نهاية التكرار" (End of loop). To the right of these fields is a button labeled "For Loop". Below these elements is a large list box labeled "listBox1" in its top right corner. The list box is currently empty.

بالضغط مرتين بالماوس على Button Double ينشئ لنا

الحدث Button1\_Click ()

قم بكتابة الكود التالي بداخله كالتالى :

```
private void button1_Click(object sender, EventArgs e)
{
    listBox1.Items.Clear();

    int answer = 0;

    int loopStart = int.Parse(textBox1.Text);
    int loopEnd = int.Parse(textBox2.Text);

    for (int i = loopStart; i <= loopEnd; i++)
    {
        answer = answer + i;
        listBox1.Items.Add(answer.ToString());
    }
}
```

### برنامج (6) تدريب عملي

أنشئ مشروع جديد من قائمة File -> New -> Project من فيجوال ستوديو

واختار نوع المشروع Windows Forms Application باسم

For Loop Total Number لإظهار مجموع الأرقام من 1 إلى 100 وإظهار

النتائج فى MessageBox و Label

ضع على الفورم التى تظهر لك الأدوات التالية وأعطى لها الخصائص الآتية:

#### 1- Button:

Name: button1

Text: مجموع الأعداد من 1-100

Font Name: Times New Roman, 14pt, style=Bold

Size: 14

Bold: True

## 2- Label:

Name: label2

Font Name: Times New Roman, 14pt, style=Bold

Size: 14

Bold: True



بالضغط مرتين بالماوس على Button Double ينشئ لنا

الحدث Button1\_Click ()

قم بكتابة الكود التالي بداخله كالتالي :

```

        private void button1_Click(object sender, EventArgs e)
        {
            int answer = 0;

            for (int i = 0; i < 101; i++)
            {
                answer = answer + 1;

                label11.Text = "الناتج =" + answer.ToString ();
            }

            MessageBox.Show("الناتج =" + answer.ToString());
        }

```

## 2- جملة () While :

تقوم هذه الجملة بتكرار عدد معين من الجمل بناءً على عدد معين يتم تحديده داخل شرط الجملة مع وضع عداد للجمل بين أقواس الجملة **While{}**

## برنامج (7)

أنشئ مشروع جديد من قائمة **File -> New -> Project** من فيجوال ستوديو واختر نوع المشروع **Windows Forms Application** باسم

## **While Loop**

ضع على الفورم التي تظهر لك الأدوات التالية وأعطى لها الخصائص الآتية:

## 1- Button:

Name: button1

Text: While Loop

Font Name: Times New Roman, 14pt, style=Bold

Size: 14

Bold: True

## 2- Textbox:

Name: textBox1

Font Name: Times New Roman, 14pt, style=Bold

Size: 14

Bold: True

Right-to-left: Yes

## 3- Textbox:

Name: textBox2

Right-to-left: Yes

Font Name: Times New Roman, 14pt, style=Bold

Size: 14

Bold: True

## 4- Label:

Name: label1

Text : بداية التكرار

Font Name: Times New Roman, 14pt, style=Bold

**Size: 14**

**Bold: True**

#### **4- Label:**

**Name: label2**

**Text :** نهاية التكرار

**Font Name: Times New Roman, 14pt, style=Bold**

**Size: 14**

**Bold: True**

#### **5- List box:**

**Name: listBox1**

**Right-to-left: Yes**

**Font Name: Times New Roman, 14pt, style=Bold**

**Size: 14**

**Bold: True**

The screenshot shows a Windows application window titled "Form1". Inside the window, there are two text input fields. The first field is labeled "بداية التكرار" (Start of loop) and the second is labeled "نهاية التكرار" (End of loop). To the right of these fields is a button labeled "While Loop". Below these elements is a large list box, which is currently empty, and it is labeled "listBox1" in its top right corner. The text is in Arabic and the font appears to be Times New Roman.

بالضغط مرتين بالماوس على Button Double ينشئ لنا

الحدث Button1\_Click ()

قم بكتابة الكود التالي بداخله كالتالى :

```
private void button1_Click(object sender, EventArgs e)
{
    int loopStart = int.Parse(textBox1.Text);
    int loopEnd = int.Parse(textBox2.Text);

    int i = 0;
    int answer = 0;
    while (i < loopEnd )
    {
        answer = answer + i;
        listBox1.Items.Add(answer.ToString());

        i++;
    }
}
```

### 3- جملة While();do{ ....}

تقوم هذه الجملة بتكرار عدد معين من الجمل بناءً على عدد معين يتم تحديده داخل شرط الجملة مع وضع عداد للجمل بين أقواس الجملة do While(); والفرق بينها وبين جملة While() أنه فى جملة do{}While() يتم تنفيذ جمل الكود بداخل do على الأقل مرة واحدة قبل تنفيذ الشرط .

### برنامج (8) تدريب عملي

أنشئ مشروع جديد من قائمة File -> New -> Project من فيجوال ستوديو واختر نوع المشروع Windows Forms Application باسم

DoWhileLoop

ضع على الفورم التى تظهر لك الأدوات التالية وأعطى لها الخصائص الآتية:

### 1- Button:

Name: button1

Text:do While Loop

Font Name: Times New Roman, 14pt, style=Bold

Size: 14

Bold: True

### 2- Textbox:

Name: textBox1

Font Name: Times New Roman, 14pt, style=Bold

Size: 14

Bold: True

Right-to-left: Yes

### 3- Textbox:

Name: textBox2

Right-to-left: Yes

Font Name: Times New Roman, 14pt, style=Bold

Size: 14

Bold: True

### 4- Label:



**Name: label1**

**Text :** بداية التكرار

**Font Name: Times New Roman, 14pt, style=Bold**

**Size: 14**

**Bold: True**

#### **4- Label:**

**Name: label2**

**Text :** نهاية التكرار

**Font Name: Times New Roman, 14pt, style=Bold**

**Size: 14**

**Bold: True**

#### **5- List box:**

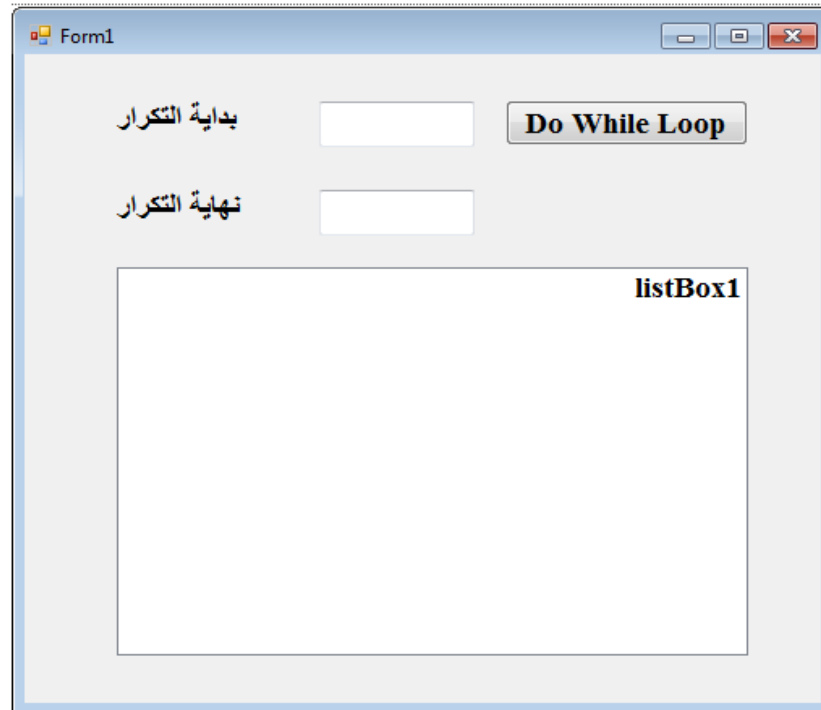
**Name: listBox1**

**Right-to-left: Yes**

**Font Name: Times New Roman, 14pt, style=Bold**

**Size: 14**

**Bold: True**



بالضغط مرتين بالماوس على Button Double ينشئ لنا

الحدث Button1\_Click ()

قم بكتابة الكود التالي بداخله كالتالي :

```
private void button1_Click(object sender, EventArgs e)
{
    int loopStart = int.Parse(textBox1.Text);
    int loopEnd = int.Parse(textBox2.Text);

    int i = 0;
    int answer = 0;
    do
    {
        answer = answer + i;
        listBox1.Items.Add(answer.ToString());

        i++;
    } while (i < loopEnd);
}
```

## الدوال (5) Methods

الدوال تقوم بتنفيذ مجموعة من سطور الكود داخل الأقواس {} ويمكن أن ترجع قيمة كانت نصية أو رقمية ويمكن أن نكتب عدد من المدخلات تسمى Parameters أو نتركها بدون Parameters ويمكن للقيمة أن نضعها في متغير وصيغة الدوال هي كما يلي :

**(Type)MethodName(Param1 , Param2 , .....ParamN)**

```
{  
  
    Statements;  
  
    Return ;  
  
}
```

### برنامج (9) تدريب عملي

أنشئ مشروع جديد من قائمة File -> New -> Project من فيجوال ستوديو واختر نوع المشروع Windows Forms Application باسم Methods ضع على الفورم التي تظهر لك الأدوات التالية وأعطى لها الخصائص الآتية:

#### 1- Label:

Name: label1

Text : الرقم الأول

Font Name: Times New Roman, 14pt, style=Bold

Size: 14

**Bold: True**

## **2- Label:**

**Name: label2**

**Text :** الرقمائى

**Font Name: Times New Roman, 14pt, style=Bold**

**Size: 14**

**Bold: True**

## **3- Label:**

**Name: lblResult**

**Font Name: Times New Roman, 14pt, style=Bold**

**Size: 14**

**Bold: True**

## **4- Textbox:**

**Name: textBox1**

**Font Name: Times New Roman, 14pt, style=Bold**

**Size: 14**

**Bold: True**

**Right-to-left: Yes**

## **5- Textbox:**

**Name: textBox2**

**Right-to-left: Yes**

**Font Name: Times New Roman, 14pt, style=Bold**

**Size: 14**

**Bold: True**

## **6- Button:**

**Name: button1**

**Text:** جمع

**Font Name: Times New Roman, 14pt, style=Bold**

**Size: 14**

**Bold: True**

## **7- Button:**

**Name: button1**

**Text:** طرح

**Font Name: Times New Roman, 14pt, style=Bold**

**Size: 14**

**Bold: True**

## **8- Button:**

**Name: button1**

**Text:** ضرب

**Font Name: Times New Roman, 14pt, style=Bold**

**Size: 14**

**Bold: True**

## **9- Button:**

**Name: button1**

**Text:**قسمة

**Font Name:** Times New Roman, 14pt, style=Bold

**Size:** 14

**Bold:** True



**كتابة الدوال التالية لإستدعاءها فى كل Button :**

```
// Add Method
private int Add(int Num1, int Num2)
{
    int answer = 0;

    answer = Num1 + Num2;

    return answer;
}

// Subtract Method
private int Subtract(int Num1, int Num2)
{
    int answer = 0;

    answer = Num1 - Num2;

    return answer;
}

// Multiply Method
```

```

private int Multiply(int Num1, int Num2)
{
    int answer = 0;

    answer = Num1 * Num2;

    return answer;
}

// Divide Method
private int Divide(int Num1, int Num2)
{
    int answer = 0;

    answer = Num1 / Num2;

    return answer;
}

// Clear Method
private void Clear()
{
    textBox1.Text = String.Empty;
    textBox2.Text = "";
    lblResult.Text = "";
}

```

بالضغط مرتين بالماوس على Button Double ينشئ لنا

الحدث Button1\_Click ()

قم بكتابة الكود التالي بداخله كالتالي :

```

private void button1_Click(object sender, EventArgs e)
{
    int Number1 = int.Parse(textBox1.Text);
    int Number2 = int.Parse(textBox2.Text);

    int result = 0;

    result = Add(Number1 ,Number2);

    lblResult.Text = " = الناتج " + result.ToString();

}

```

بالضغط مرتين بالماوس على Button Double ينشئ لنا

الحدث Button2\_Click ()

قم بكتابة الكود التالي بداخله كالتالي :

```
private void button2_Click(object sender, EventArgs e)
{
    int Number1 = int.Parse(textBox1.Text);
    int Number2 = int.Parse(textBox2.Text);

    int result = 0;

    result = Subtract(Number1, Number2);

    lblResult.Text = "الناتج =" + result.ToString();
}
```

بالضغط مرتين بالماوس على Button Double ينشئ لنا

الحدث Button3\_Click ()

قم بكتابة الكود التالي بداخله كالتالي :

```
private void button3_Click(object sender, EventArgs e)
{
    int Number1 = int.Parse(textBox1.Text);
    int Number2 = int.Parse(textBox2.Text);

    int result = 0;

    result = Multiply(Number1, Number2);

    lblResult.Text = "الناتج =" + result.ToString();
}
```

بالضغط مرتين بالماوس على Button Double ينشئ لنا

الحدث Button4\_Click ()

قم بكتابة الكود التالي بداخله كالتالي :



```
private void button4_Click(object sender, EventArgs e)
{

    int Number1 = int.Parse(textBox1.Text);
    int Number2 = int.Parse(textBox2.Text);

    int result = 0;

    result = Divide(Number1, Number2);

    lblResult.Text = "الناتج =" + result.ToString();
}
```

## **Arrays (6) المصفوفات**

هي مجموعة من العناصر لها نفس النوع مرتبة بما يسمى Index ويبدأ بالرقم 0 ويتم تعريفها كالتالي :

**Type array\_name[position\_in\_array] = array\_value;**

وتحديد عدد One-Dimensional Arrays لتعريف المصفوفة ذات البعد الواحد العناصر التي يمكن أن تتسع لها المصفوفة

**int[ ] lottery\_numbers = new int[49];**

لإعطاء قيم للمصفوفة بطريقتين كالتالي :

(أ) الطريقة الأولى :

**Numbers [0] = 1;**

**Numbers [1] = 2;**

**Numbers [2] = 3;**

**Numbers [3] = 4;**

(ب) الطريقة الثانية :

**int[] Numbers = new int[4] { 1, 2, 3, 4 };**

**برنامج (10)**

أنشئ مشروع جديد من قائمة File -> New -> Project من فيجوال ستوديو

واختار نوع المشروع Windows Forms Application باسم

OneDimentionArray مصفوفة ذات البعد الواحد

ضع على الفورم التي تظهر لك الأدوات التالية وأعطى لها الخصائص الآتية:

## 1- Button:

Name: button1

Text:OneDimensionArray

Font Name: Times New Roman, 14pt, style=Bold

Size: 14

Bold: True

## 2- List box:

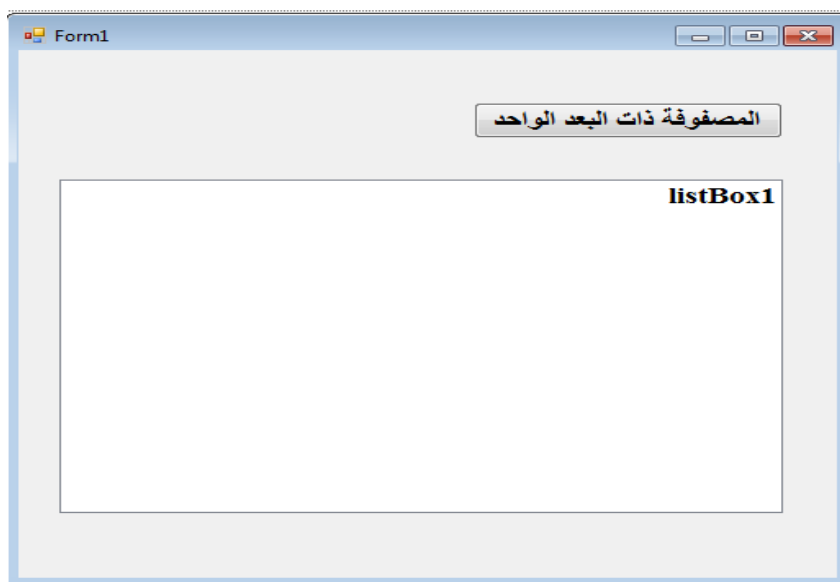
Name: listBox1

Right-to-left: Yes

Font Name: Times New Roman, 14pt, style=Bold

Size: 14

Bold: True



بالضغط مرتين بالماوس على Button Double ينشئ لنا

الحدث Button1\_Click ()

قم بكتابة الكود التالي بداخله كالتالى :

```
private void button1_Click(object sender, EventArgs e)
{
    int[] Numbers = new int[4];

    Numbers[0] = 1;
    Numbers[1] = 2;
    Numbers[2] = 3;
    Numbers[3] = 4;

    // طريقة for
    for (int i = 0; i < Numbers.Length; i++)
    {
        listBox1.Items.Add(Numbers[i]);
    }

    // طريقة foreach
    foreach (int item in Numbers)
    {
        listBox1.Items.Add(item);
    }
}
```

جملة Foreach() :

وهي مثل جملة for ولكنها ليس لها عدد محدد من العناصر بل تقوم هذه الجملة بعمل تكرار على مجموعة من العناصر داخل مصفوفة

```
foreach (type item in Collection)
{
    Statements;
}
```

## تعريف مصفوفة ذات البعدين : Two-Dimensional Arrays:

(أ) الطريقة الأولى :

`String [ , ] Names ;`

(إ) الطريقة الثانية :

`String [ ][ ] Names ;`

|       | Column 0                 | Column 1                 | Column 2                 | Column 3                 |
|-------|--------------------------|--------------------------|--------------------------|--------------------------|
| Row 0 | <code>a[ 0 ][ 0 ]</code> | <code>a[ 0 ][ 1 ]</code> | <code>a[ 0 ][ 2 ]</code> | <code>a[ 0 ][ 3 ]</code> |
| Row 1 | <code>a[ 1 ][ 0 ]</code> | <code>a[ 1 ][ 1 ]</code> | <code>a[ 1 ][ 2 ]</code> | <code>a[ 1 ][ 3 ]</code> |
| Row 2 | <code>a[ 2 ][ 0 ]</code> | <code>a[ 2 ][ 1 ]</code> | <code>a[ 2 ][ 2 ]</code> | <code>a[ 2 ][ 3 ]</code> |

```
int[, ] a =int[3,4]={  
{0,1,2,3},/* initializers for row indexed by 0 */  
{4,5,6,7},/* initializers for row indexed by 1 */  
{8,9,10,11}/* initializers for row indexed by 2 */  
};
```

وللوصول لعنصر في المصفوفة يتم بطريقة Subscript وهو معرفة

Row index & Column index of the Array كالتالى :

```
int val = a[2,3];
```

## تعريف مصفوفة ذات ثلاثة أبعاد : Three-Dimensional Arrays:

(ب) الطريقة الأولى :

`String [ , , ] Names ;`

(ب) الطريقة الثانية :

`String [ ][ ][ ] Names ;`

وهكذا تتم تعريف المصفوفات كلما أرت زيادة البعد لها

## برنامج (11)

أنشئ مشروع جديد من قائمة **File -> New -> Project** من فيجوال ستوديو

واختار نوع المشروع **Windows Forms Application** باسم

**TwoDimentionArray** مصفوفة ذات البعدين

ضع على الفورم التي تظهر لك الأدوات التالية وأعطى لها الخصائص الآتية:

### **1- Button:**

**Name:** button1

**Text:**TwoDimensionArray

**Font Name:** Times New Roman, 14pt, style=Bold

**Size:** 14

**Bold:** True

### **2- Rich Textbox:**

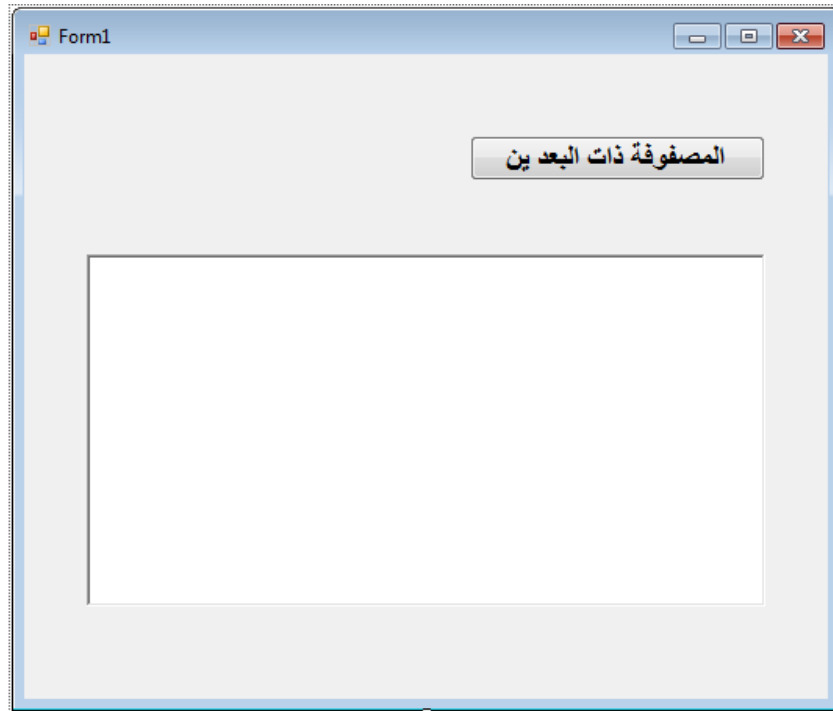
**Name:** richTextBox1

**Right-to-left:** Yes

**Font Name:** Times New Roman, 14pt, style=Bold

**Size:** 14

**Bold:** True



بالضغط مرتين بالماوس على Button Double ينشئ لنا

الحدث Button1\_Click ()

قم بكتابة الكود التالي بداخله كالتالي :

```
private void button1_Click(object sender, EventArgs e)
{
    /* an array with 5 rows and 2 columns*/
    int[,] a = new int[5, 2] { { 0, 0 }, { 1, 2 }, { 2, 4 }, { 3, 6 }, { 4, 8 } };

    int i, j;

    /* output each array element's value */
    for (i = 0; i < 5; i++)
    {
        for (j = 0; j < 2; j++)
        {

        }
    }

    //Subscripting

    // 0 0
    // 1 2
    // 2 4
    // 3 6
    // 4 8
}
```

```
// |  
// v
```

```
//a[0,0] = 0  
//a[0,1] = 0  
//a[1,0] = 1  
//a[1,1] = 2  
//a[2,0] = 2  
//a[2,1] = 4  
//a[3,0] = 3  
//a[3,1] = 6  
//a[4,0] = 4  
//a[4,1] = 8
```

```
richTextBox1.Text += "a["+i+", "+j+"] = " + a[i, j] + "\n";
```

```
    }  
}  
}
```



## Events الأحداث (7)

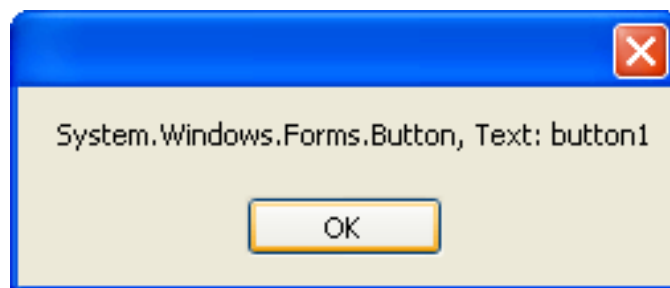
### 1 – حدث Click

انقر فوق الحدث للزر:

```
private void button1_Click(object sender, EventArgs e)
{
}
}
```

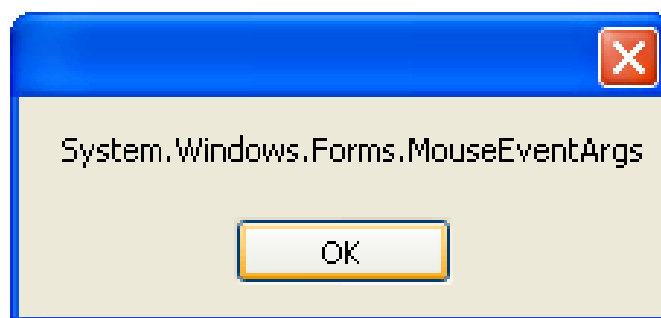
تشغيل البرنامج وانقر فوق الزر. يجب أن تشاهد هذا:

```
MessageBox.Show( sender.ToString() );
```



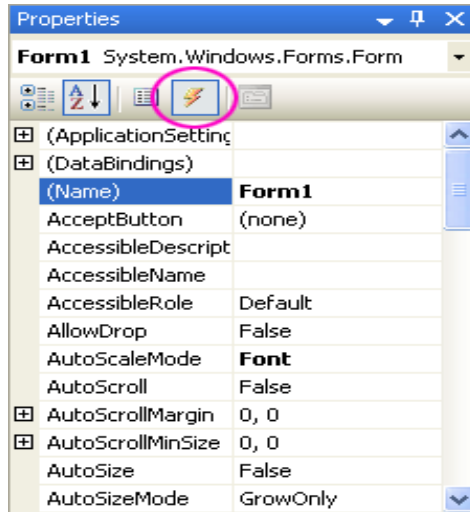
```
MessageBox.Show( e.ToString());
```

مربع النص سوف يظهر ما يلي :

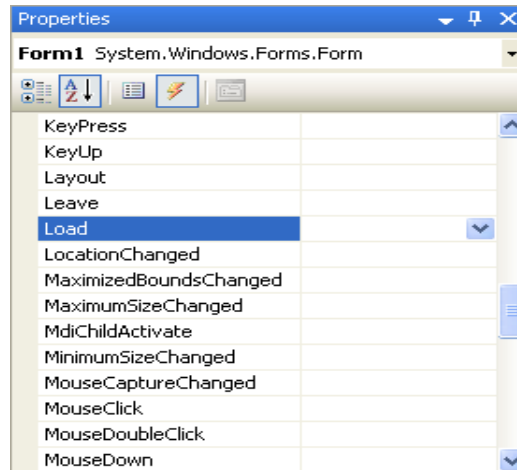


## The MouseDown Event in C# .NET - 2

الخصائص :لنربما هي الأحداث المتاحة للنموذج نفسه، انقر فوق الجزء العلوي من قائمة الخصائص، كما في الصورة أدناه،



عند الضغط على الجزء العلوي الأصفر في قائمة الخصائص ستظهر الأحداث كما يلي :



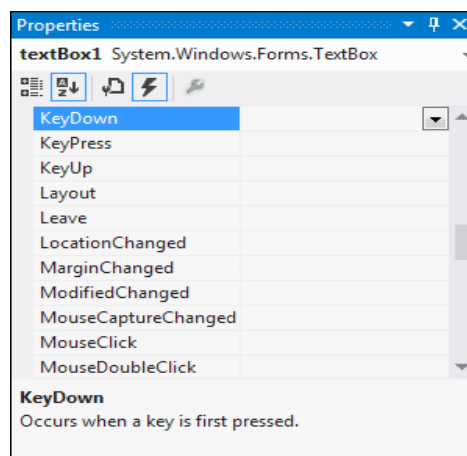
كما يمكن استخدام جملة if لاختبار أي من ازرار الماوس تقوم باختياره كما يلي :

```
private void Form1_MouseDown(object sender, MouseEventArgs e)
{
    if (e.Button == MouseButtons.Left)
    {
        MessageBox.Show("Left button clicked");
    }
    else if (e.Button == MouseButtons.Right)
    {
        MessageBox.Show("Right button clicked");
    }
}
```

عند تشغيل البرنامج والنقر فوق أي من أزرار الماوس على النموذج. يجب أن تشاهد عرض مربع رسالة.

## The KeyDown Event in C# .NET - 3

من قائمة الخصائص حدد حدث KeyDown لمربع النص الخاص :

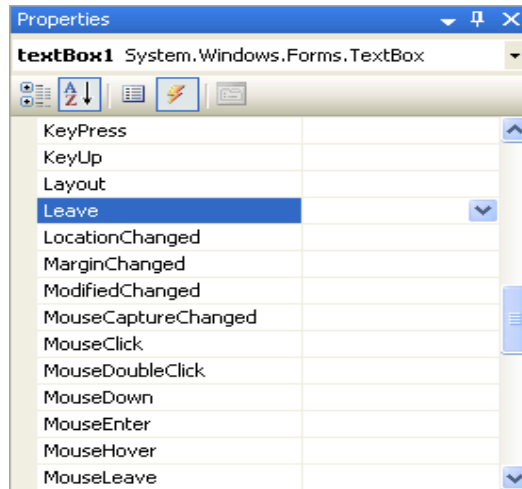


```
private void Form1_KeyDown(object sender, KeyEventArgs e)
{
    |

    if (e.KeyData == Keys.A)
    {
        MessageBox.Show("Letter A Pressed");
    }
}
```

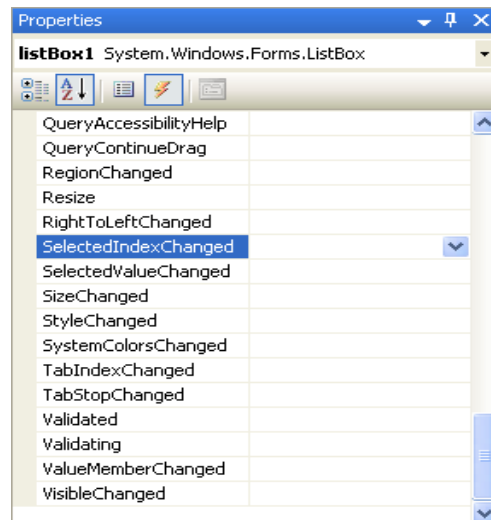
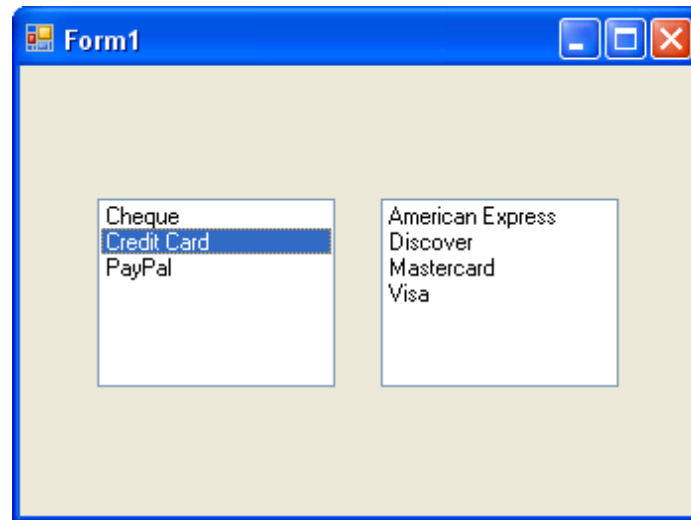
## The Leave Event in C# .NET - 4

هذا الحدث مفيد جدا يمكنك استخدامها لمربعات النص هو حدث Leave. فإنه يسمح لك للتحقق من صحة مربع النص عندما يحاول المستخدم إلى تركها.



```
private void textBox1_Leave(object sender, EventArgs e)
{
    if (textBox1.Text == "")
    {
        MessageBox.Show("You can't leave this box blank");
        textBox1.Focus();
    }
    else
    {
        textBox1.Text = textBox1.Text.ToUpper();
    }
}
```

## 5- أحداث ListBox ListBox Events



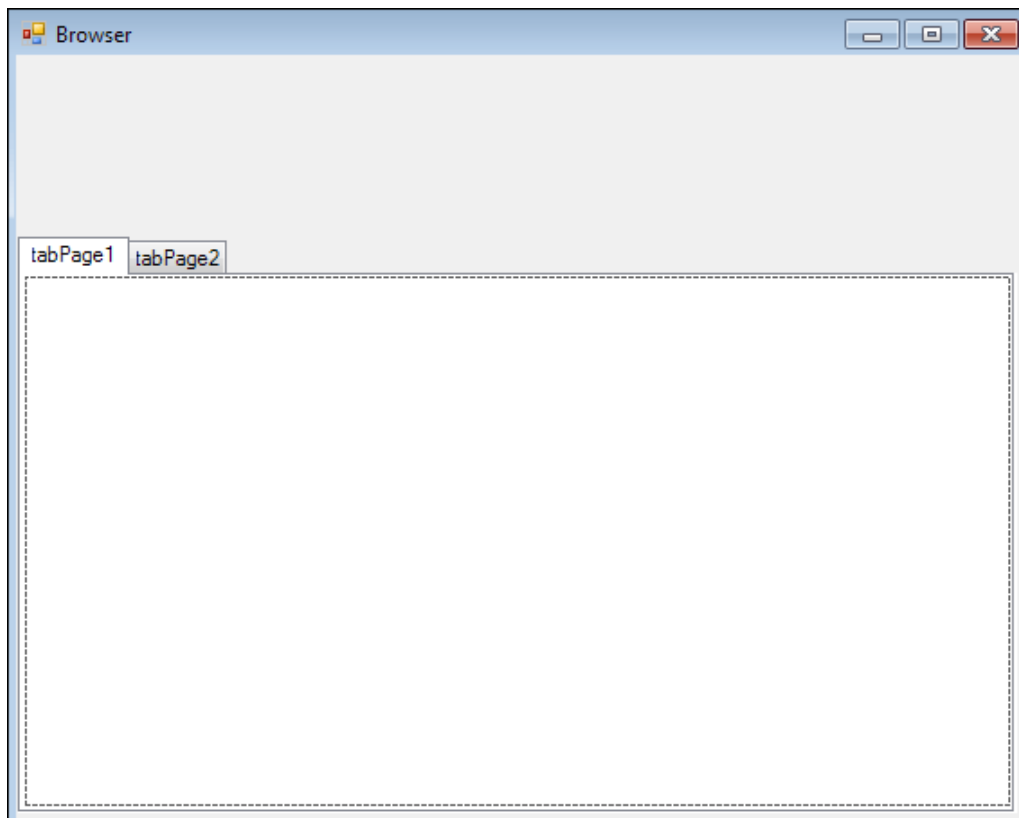
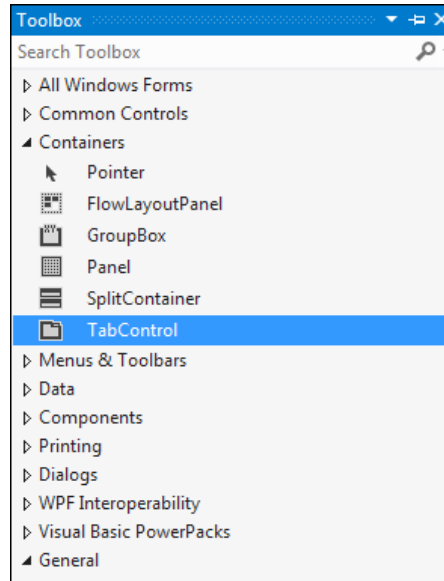
```
private void listBox1_SelectedIndexChanged(object sender, EventArgs e)
{
    if (listBox1.SelectedIndex == 1)
    {
        loadListBox();
    }
}

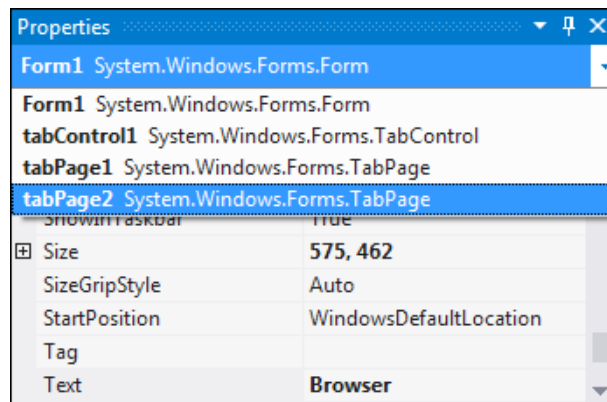
private void loadListBox()
{
    listBox2.Items.Clear();

    listBox2.Items.Add("American Express");
    listBox2.Items.Add("Discover");
    listBox2.Items.Add("Mastercard");
    listBox2.Items.Add("Visa");
}
```

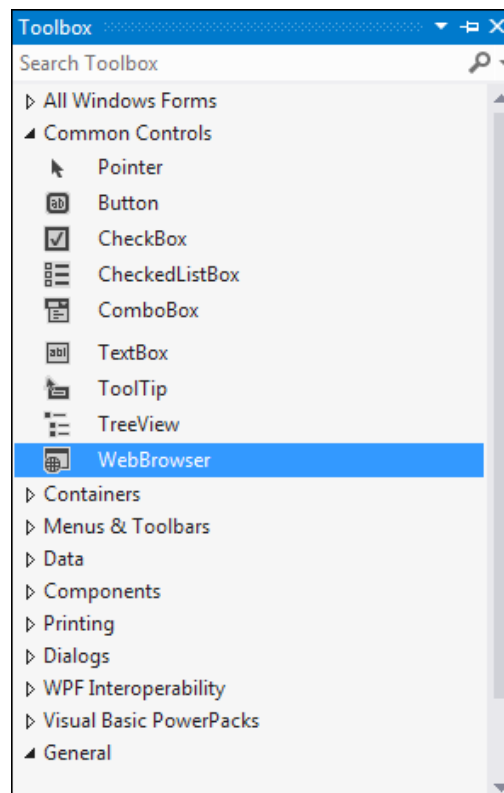
## 7 - عمل متصفح الخاص بلغة C#

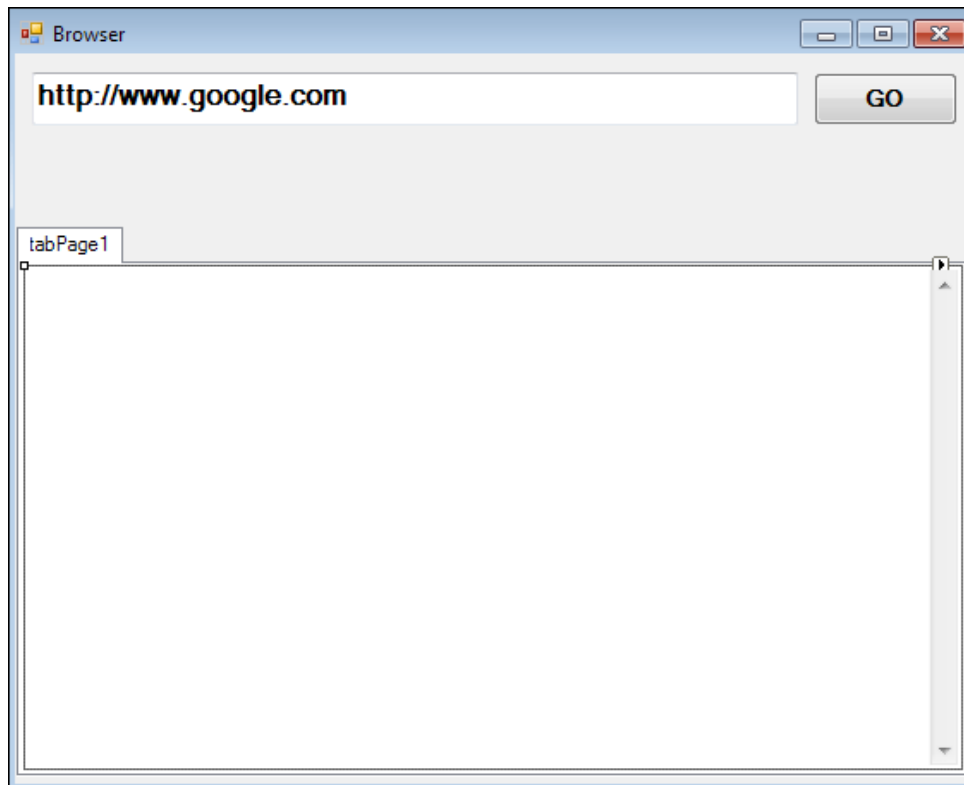
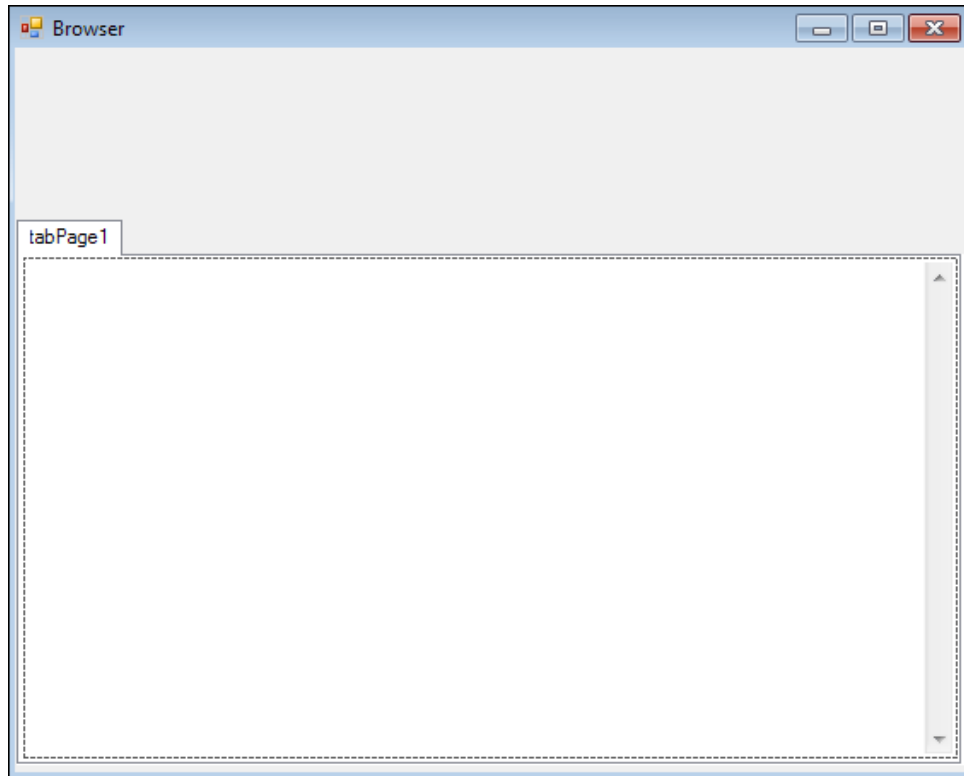
### Build your own C# Custom Web Browser





## : أداة Web Browser Control

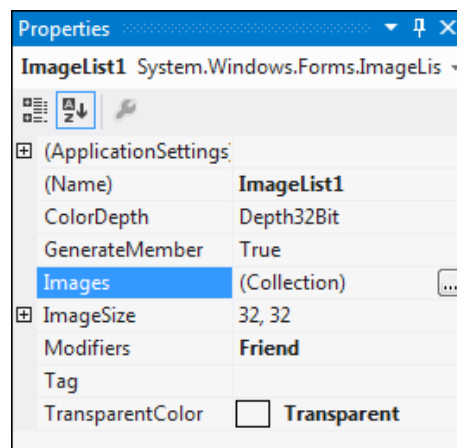
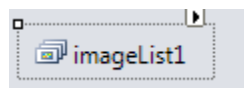
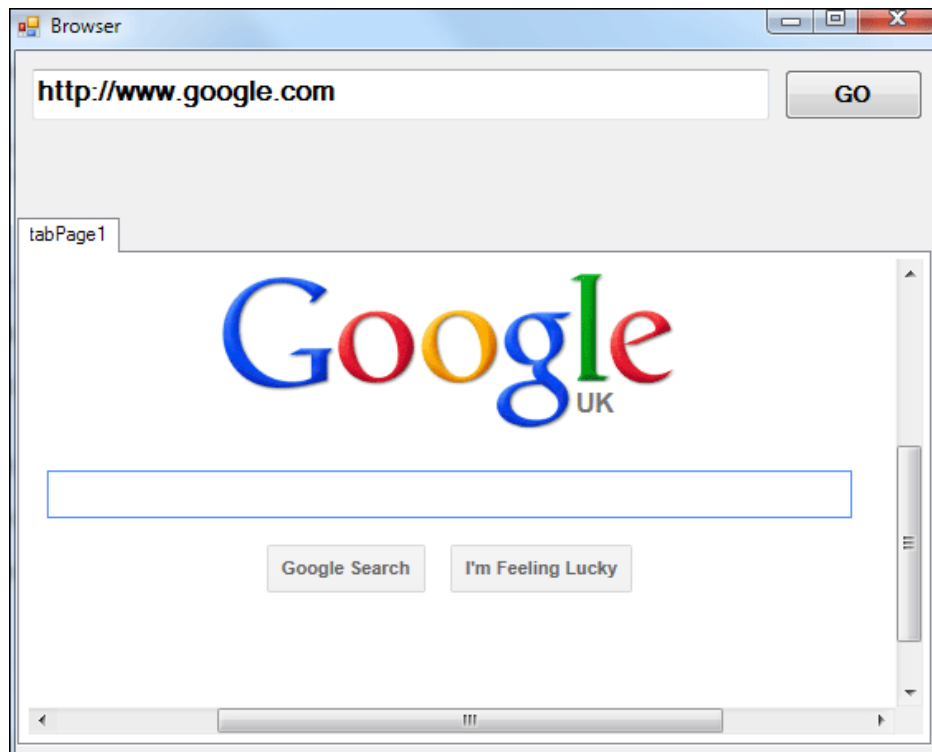


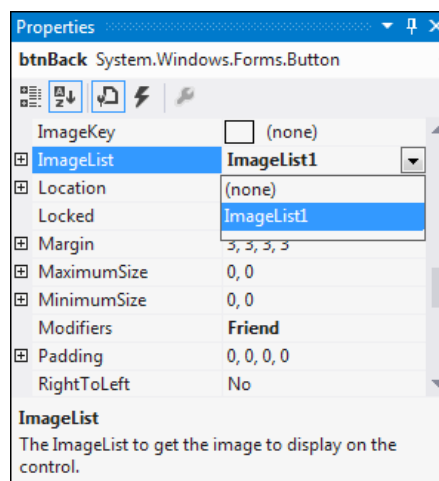
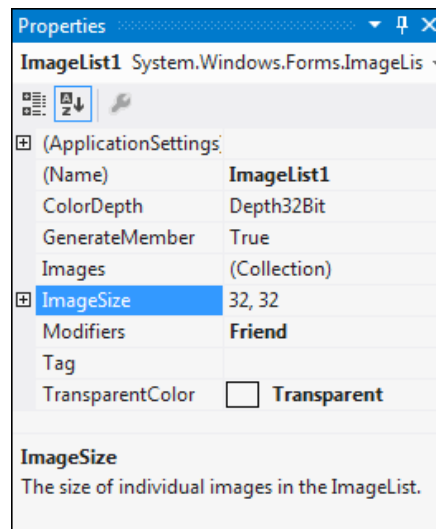
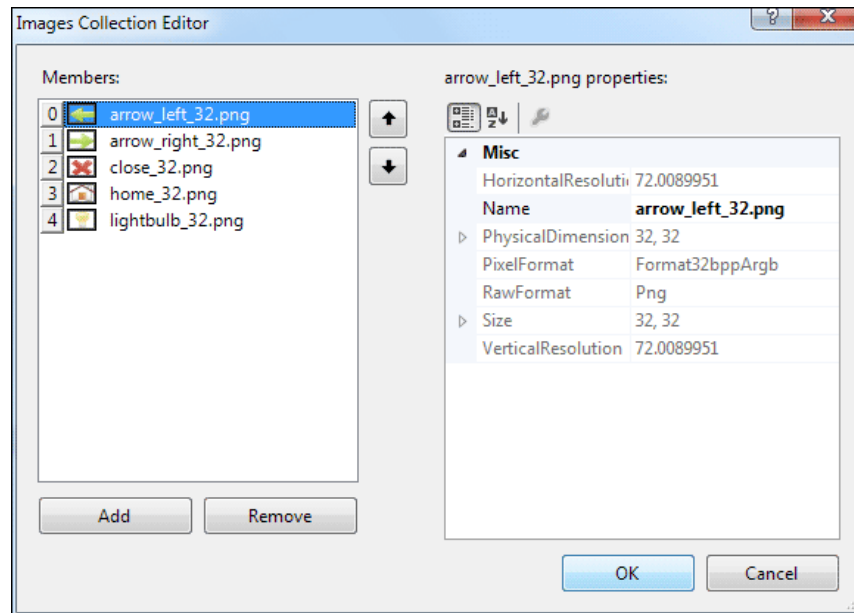


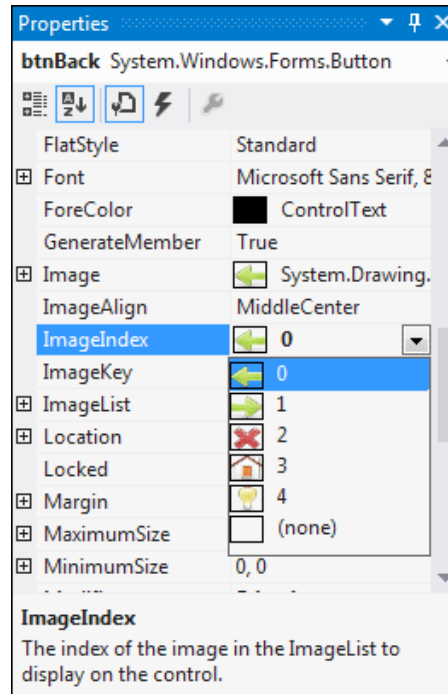


```
string WebPage = txtAddress.Text.Trim();
```

```
webBrowser1.Navigate(WebPage);
```







**کود Button Back :**

```
if (webBrowser1.CanGoBack)
{
    webBrowser1.GoBack();
}
```

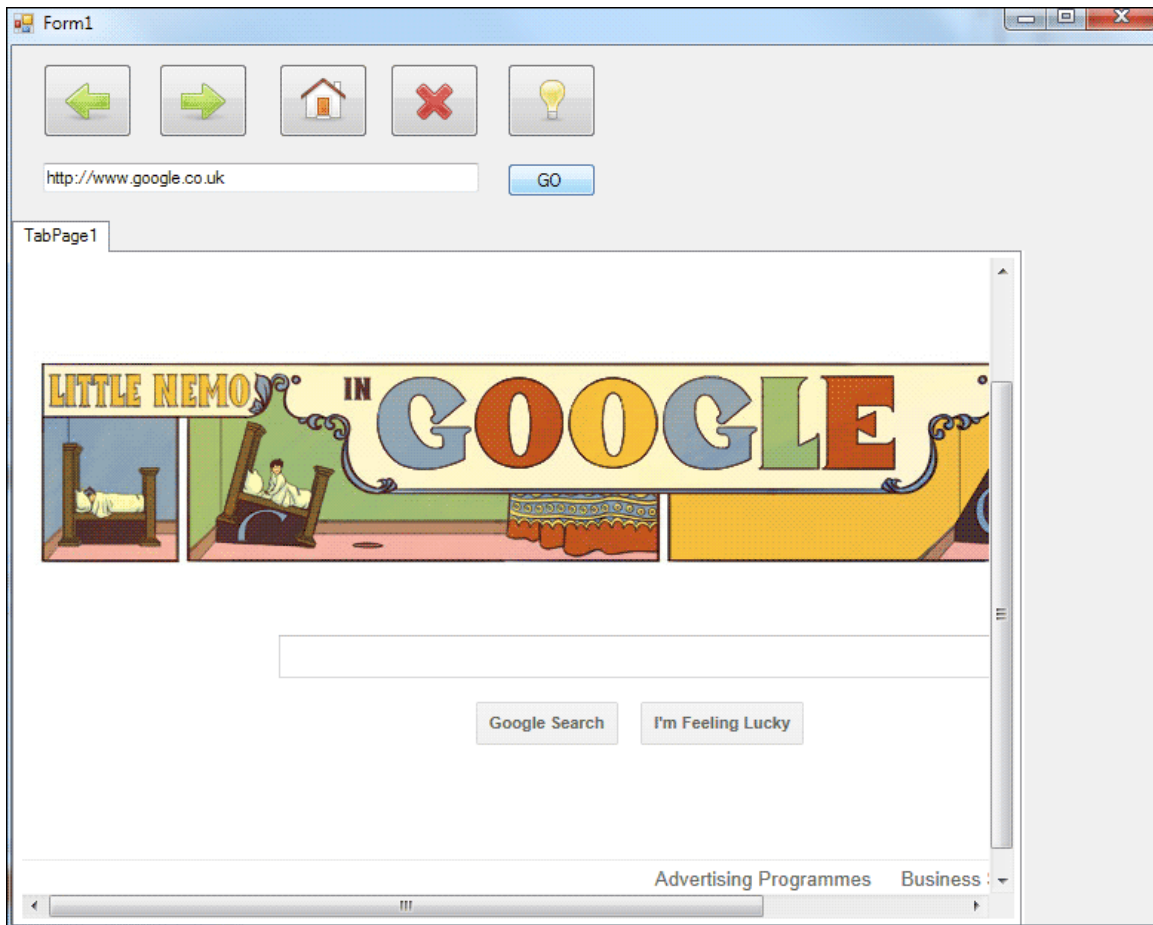
**Code for Button Forward:**

```
if (webBrowser1.CanGoForward)
{
    webBrowser1.GoForward();
}
```

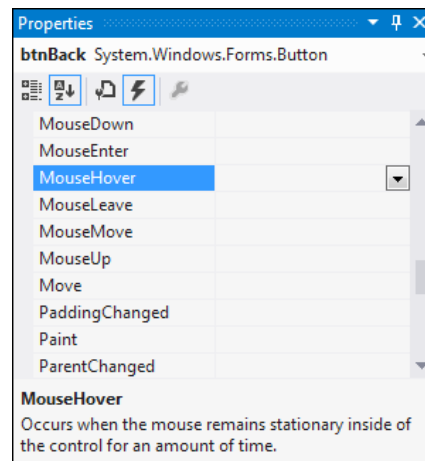
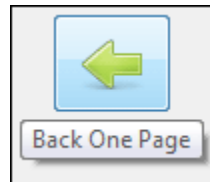
Code for Button Stop:webBrowser1.Stop( );

Code for Button Home:webBrowser1.GoHome( );

Code for Button Refresh:webBrowser1.Refresh( );



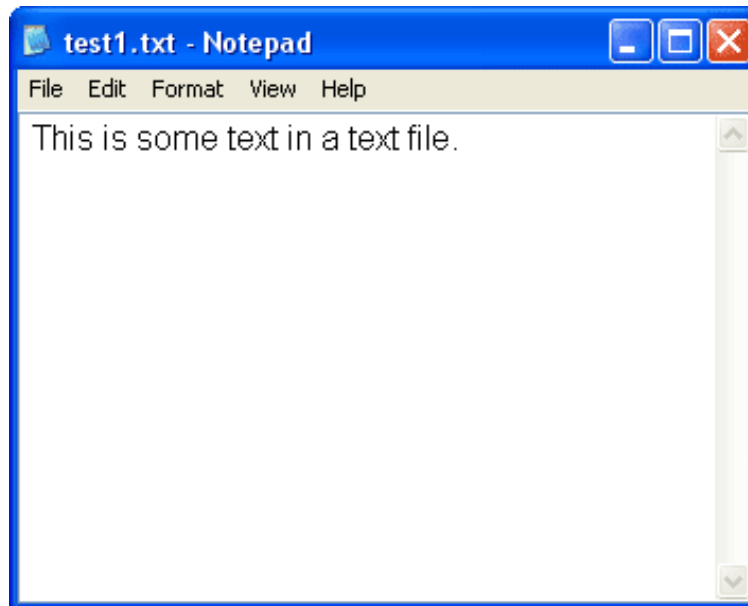
## 8-إضافة ToolTips in C# .NET



```
toolTip1.SetToolTip( btnBack, "Back One Page" );
```

## Manipulating Files التعامل مع الملفات (8)

### 1- فتح Text File in C# .NET



```
private void btnOpen_Click(object sender, EventArgs e)
{
    string file_name = "C:\\\\test1111.txt";

    if (System.IO.File.Exists(file_name) == true)
    {
        System.IO.StreamReader objReader;
        objReader = new System.IO.StreamReader(file_name);

        textBox1.Text = objReader.ReadToEnd();

        objReader.Close();
    }
    else
    {
        MessageBox.Show("No such file " + file_name);
    }
}
```

## 2- قراءة ملف سطر سطر

```
private void btnOpen_Click(object sender, EventArgs e)
{
    string file_name = "C:\\test1.txt";
    string textLine = "";

    System.IO.StreamReader objReader;
    objReader = new System.IO.StreamReader(file_name);

    do
    {
        textLine = textLine + objReader.ReadLine() + "\r\n";
    } while (objReader.Peek() != -1);

    textBox1.Text = textLine;

    objReader.Close();
}
```

الدالة **Peek** تختبر ما إذا كان هناك حرف معين في الاختبار أم لا إذا لم يكن هناك حرف فإنها تسترجع القيمة -1 .

## 3 الكتابة لملف Text File in C#

```
private void btnWrite_Click(object sender, EventArgs e)
{
    string file_name = "C:\\test1.txt";

    System.IO.StreamWriter objWriter;
    objWriter = new System.IO.StreamWriter(file_name);

    objWriter.Write(textBox1.Text);
    objWriter.Close();

    MessageBox.Show("Wrote File");
}
```

## 4 لنسخ ملف باستخدام C#

لنسخ ملف يمكن أن نستخدم الدالة **Copy** التابعة للفصيلة **File Class**

**System.IO.File.Copy( fileToCopy, newLocation );**

```
private void btnCopy_Click(object sender, EventArgs e)
{
    string fileToCopy = "C:\\test1.txt";
    string newLocation = "C:\\copiedFiles\\test1.txt";
    string folderLocation = "C:\\copiedFiles";

    if (System.IO.Directory.Exists(folderLocation))
    {
        if (System.IO.File.Exists(fileToCopy))
        {
            System.IO.File.Copy(fileToCopy, newLocation, true);
            MessageBox.Show("File Copied");
        }
        else
        {
            MessageBox.Show("No such File");
        }
    }
    else
    {
        MessageBox.Show("No such Directory");
    }
}
```

## 5 لنقل ملف باستخدام C#

لنقل ملف يمكن أن نستخدم الدالة **Move** التابعة للفصيلة **File Class** لمكان جديد

**System.IO.File.Move( fileToMove, fileLocation );**

## 6 لحذف ملف باستخدام C#

لحذف ملف يمكن أن نستخدم الدالة **Delete** التابعة للفصيلة **File Class** لمكان جديد

**System.IO.File.Delete( fileToMove, fileLocation );**



# Debugging (9) تصحيح أخطاء التطبيق

## Errors at Design-Time 1

### Blue Wavy Lines

```
textBox2.Text = x;
```

The name 'textBox2' does not exist in the current context

### Red Wavy Lines

```
textBox1.Text = x
```

; expected

### Green Wavy Lines

```
private void button1_Click(object sender, EventArgs e)
{
```

```
    string CompilerWarning = "";
```

The variable 'CompilerWarning' is assigned but its value is never used

```
    textBox1.Text = "";
```

```
}
```

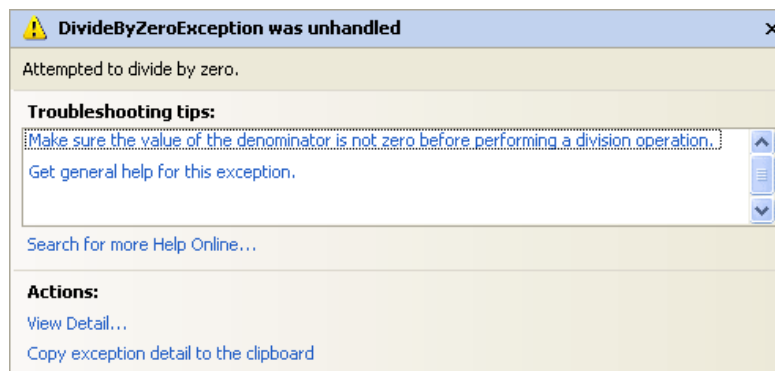
## Run-Time Error -2

```
private void button1_Click(object sender, EventArgs e)
{

    int Num1 = 10;
    int Num2 = 0;
    int answer;

    answer = Num1 / Num2;

}
```



## Logic Errors -3

Logic errors are ones where you don't get the result you were expecting.

```
private void button1_Click(object sender, EventArgs e)
{

    int startLoop = 11;
    int endLoop = 1;
    int answer = 0;

    for (int i = startLoop; i < endLoop; i++)
    {
        answer = answer + i;
    }

    MessageBox.Show("answer =" + answer.ToString());

}
```

## Breakpoints in c# -4

```
18 private void button1_Click(object sender, EventArgs e)
19 {
20     int LetterCount = 0;
21     string strText = "Debugging";
22     string letter;
23
24     for (int i = 0; i < strText.Length; i++)
25     {
26         letter = strText.Substring(i, i + 1);
27
28         if (letter == "g")
29         {
30             LetterCount++;
31         }
32     }
33
34     textBox1.Text = "g appears " + LetterCount + " times";
35 }
```

```
18 private void button1_Click(object sender, EventArgs e)
19 {
20     int LetterCount = 0;
21     string strText = "Debugging";
22     string letter;
23
24     for (int i = 0; i < strText.Length; i++)
25     {
26         letter = strText.Substring(i, i + 1);
27
28         if (letter == "g")
29         {
30             LetterCount++;
31         }
32     }
33
34     textBox1.Text = "g appears " + LetterCount + " times";
35 }
```

```

18 private void button1_Click(object sender, EventArgs e)
19 {
20     int LetterCount = 0;
21     string strText = "Debugging";
22     string letter;
23
24     for (int i = 0; i < strText.Length; i++)
25     {
26         letter = strText.Substring(i, i + 1);
27
28         if (letter == "g")
29         {
30             LetterCount++;
31         }
32     }
33
34     textBox1.Text = "A appears " + LetterCount + " times";

```

```

for (int i = 0; i < strText.Length; i++)
{
    letter = strText.Substring(i, i + 1);
    letter = "e"
    if (letter == "g")
    {
        LetterCount++;
    }
}

```

```

for (int i = 0; i < strText.Length; i++)
{
    letter = strText.Substring(i, i + 1);
    strText = "Debugging"
    if (letter == "g")
    {
        LetterCount++;
    }
}

```

## Local window in c# -5

### اختر من القائمة Windows > Locals

```
for (int i = 0; i < strText.Length; i++)
{
    letter = strText.Substring(0, 1);

    if (letter == "g")
    {
        LetterCount++;
    }
}
```

| Locals      |                                |  |
|-------------|--------------------------------|--|
| Name        | Value                          |  |
| this        | {debugging.Form1, Text: Form1} |  |
| sender      | {Text = "button1"}             |  |
| e           | {X = 33 Y = 13 Button = Left}  |  |
| i           | 0                              |  |
| LetterCount | 0                              |  |
| strText     | "Debugging"                    |  |
| letter      | "D"                            |  |

|            |                  |            |        |       |
|------------|------------------|------------|--------|-------|
| Call Stack | Immediate Window | Error List | Locals | Watch |
|------------|------------------|------------|--------|-------|

### تلاحظ أن قيمة المتغير تتغير F10 اضغط على مفتاح

```
for (int i = 0; i < strText.Length; i++)
{
    letter = strText.Substring(0, 1);

    if (letter == "g")
    {
        LetterCount++;
    }
}
```

| Locals      |                                |  |
|-------------|--------------------------------|--|
| Name        | Value                          |  |
| this        | {debugging.Form1, Text: Form1} |  |
| sender      | {Text = "button1"}             |  |
| e           | {X = 33 Y = 13 Button = Left}  |  |
| i           | 3                              |  |
| LetterCount | 0                              |  |
| strText     | "Debugging"                    |  |
| letter      | "D"                            |  |

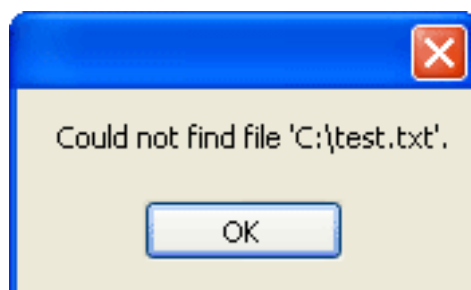
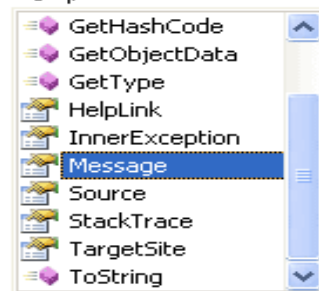
|            |                  |            |        |       |
|------------|------------------|------------|--------|-------|
| Call Stack | Immediate Window | Error List | Locals | Watch |
|------------|------------------|------------|--------|-------|

## 6 – استخدام try .... Catch in C#

يتم استخدامها للتعامل مع الأخطاء المحتملة بوضع الكود في try ويمكن تصيد الأخطاء باستخدام Catch بالكود التالي :

```
try  
{  
  
}  
catch  
{  
  
}
```

```
try  
{  
    rtb.LoadFile("C:/test.txt");  
}  
catch (System.Exception excep)  
{  
    MessageBox.Show(excep.);  
}
```



إذا كنت تعرف الخطأ الناتج يمكن استخدام الكود التالي :

```
catch (System.IO.FileNotFoundException)
{
    MessageBox.Show("File not found");
}
```

لمعرفة نوع الخطأ الناتج يمكن استخدام الكود التالي :

```
catch (System.Exception excep)
{
    MessageBox.Show( excep.GetType().ToString() );
}
```

## C# التاريخ والوقت فى لغة C# (10)

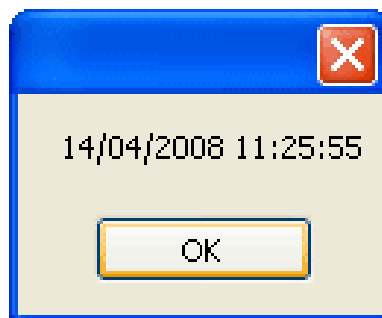
### Dates and Times in C#

Add this to you button code:

```
DateTime theDate;
```

```
theDate = DateTime.Now;
```

```
MessageBox.Show( theDate.ToString() );
```



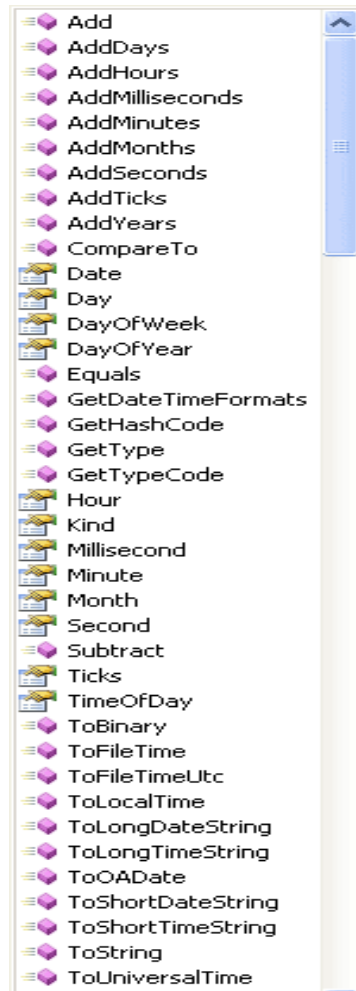
You can also have this, instead of Now:

```
theDate = DateTime.Today;
```

And even this

```
theDate = DateTime.UtcNow;
```





To use this, then, you can set up a new integer variable and hand it the day:

```
theDate = DateTime.UtcNow;
```

```
int theDay = theDate.Day;
```

The Month and Year are also Integers, so the code is similar:

```
int theMonth = theDate.Month;
```

Or

```
int theYear = theDate.Year;
```

```
DateTime firstDate = new DateTime(2008, 01, 14);
```



**// Convert To Hijri Date Method**

```
public string HijriDate(int Year, int Month, int Day)  
  
{  
  
    CultureInfo arSA = CultureInfo.CreateSpecificCulture("ar-SA");  
  
    DateTime dt = new DateTime(Year, Month, Day);  
  
    //String stringDate = "yyyy-MM-dd";  
  
    String stringDate = "dd-MM-yyyy";  
  
    string s = dt.ToString(stringDate, arSA);  
  
    return s.ToString();  
  
}
```

## برمجة الكائنات الموجهة (11)

# Object-Oriented Programming (OOP)

برمجة الكائنات الموجهة Object – Oriented Programming

لغة البرمجة تكون Full OOP

لابد وأن تتوافر بها الأربعة شروط التالية :

Encapsulation -1

Abstraction -2

Inheritance -3

Polymorphism -4

### الشرط الأول: Encapsulation

ومعناها هو تجميع الأشياء ووضعها آالة بمكانها .

ومناسمها تستطيع تمييز ذلك معناها الكبسولة

ولتعريفها جزأين :

الجزء الأول : (Class) ونضع فيها المتغيرات والدوال التي لها علاقة ببعضها

هيتعريف مكان من نمنسند إليها سماً

الجزء الثاني : هذا المكان لا يستطيع أحد الدخول إليها إلا بصلاحيات محددة من قبل المبرمج فيوجد

وهو القسم العام قسمان رئيسيان لها Public والPrivate وهو القسم الخاص

أما Protected يتم استخدامه إذا تم التعامل مع فئات مورثة

## Inherited Classes

### الشرط الثاني: Abstraction:

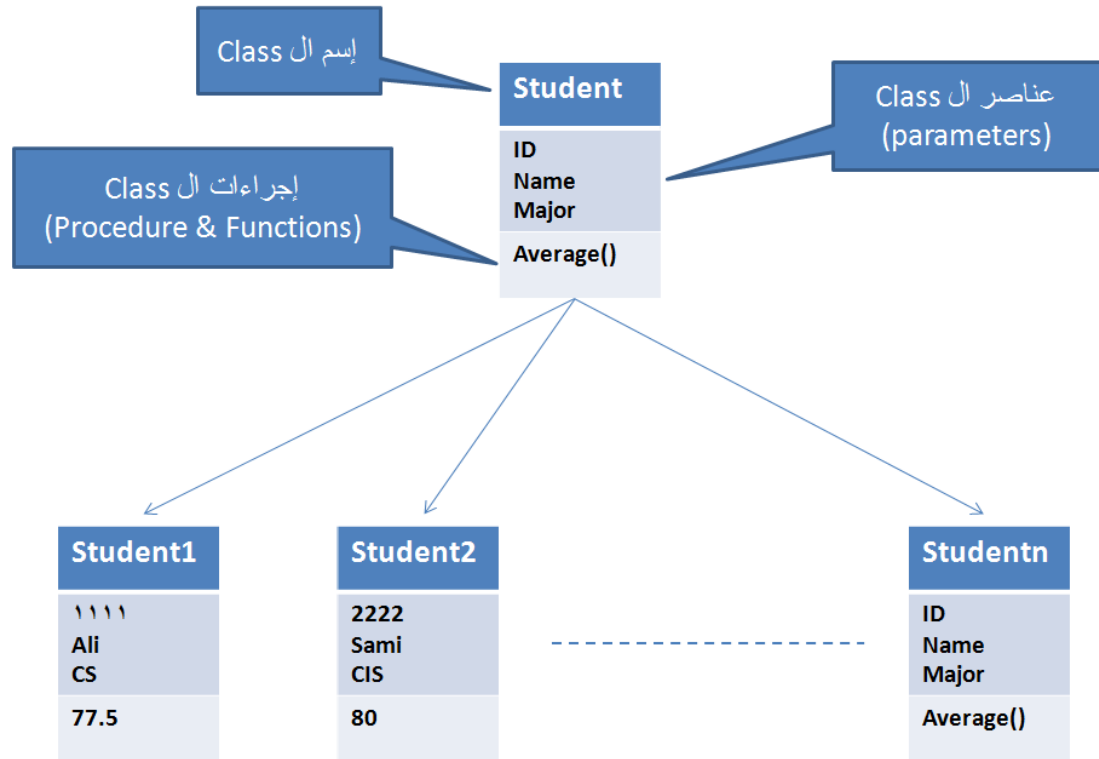
هي إضافة دالة عامة Method داخل Class لأقوم بتحديد التفاصيل لها  
إلا في قسم البرنامج الرئيسى .  
فمثلاً

(Class) أو Modules في جامعة معينة فينتج من ذلك كائن للدكتور وكائن  
نريد صنع عدة كائنات للطلاب وآخر للمدير وآخر للعاملو ... إلخ وذلك لتوفير الوقت والجهد  
عند كتابتك للكود .

### الفصيلة (Class)

هي مجموعة من النصوص البرمجية تقوم بوظيفة معينة وتقوم فكرتها على أنه  
يمكن إستخدام الكود أكثر من مرة بنفس صيغة الكود داخل مشروعك أو مشاريع  
أخرى لتحقيق مبدأ reusability أى أن نفس الكود يمكن إستخدامه أكثر من مرة  
ولا حاجة لتكرار كتابته مرة أخرى وتحتوى على خصائص ودوال ويمكن اعتبارها  
كقالب Template أو مصنع Factory يتم من خلاله انتاج ما يسمى بالهدف  
Object والذي نستخدمه للوصول والتعامل مع خصائص ودوال الفصيلة .

مثال لذلك :



## نطاق أو مجال الكود Namespace :

يعرف ببداية أى برنامج بلغة C#Namespaces المتطلبة بمجرد إنشاء البرنامج عن طريق كلمة **using** ويستخدم لتحديد أسماء الفصائل التى يتم تعريفها بداخله حتى لا يحدث خطأ أثناء إستدعاء الفصييلة كالتالى :

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace SunSchoolNameSpace
{
    class Student
    {
    }
}
  
```

```

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace MicrosoftSchoolNameSpace
{
    class Student
    {

    }

}

```

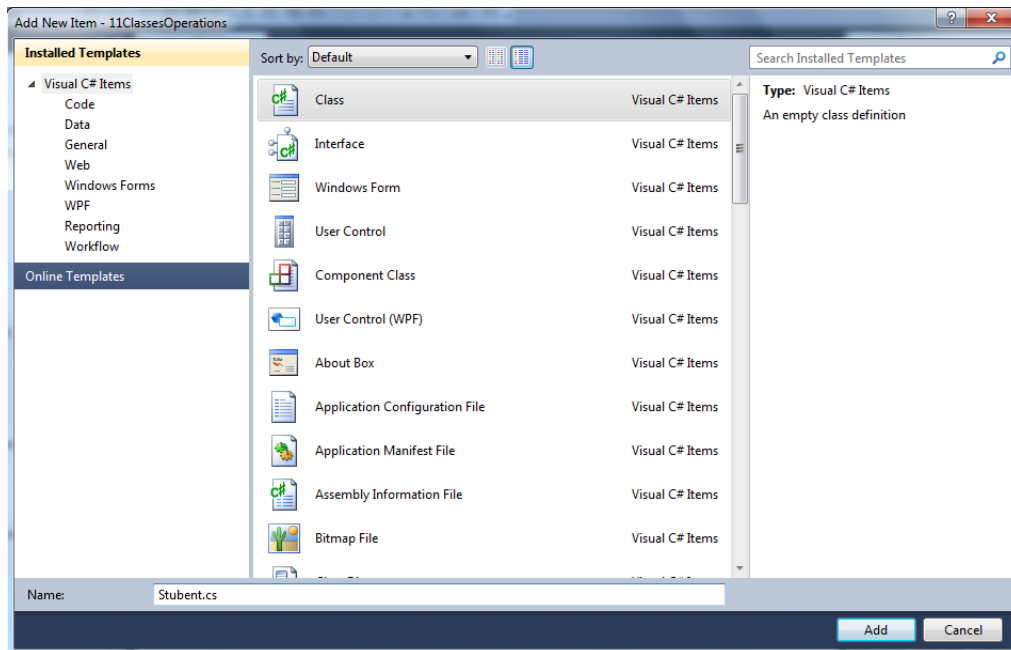
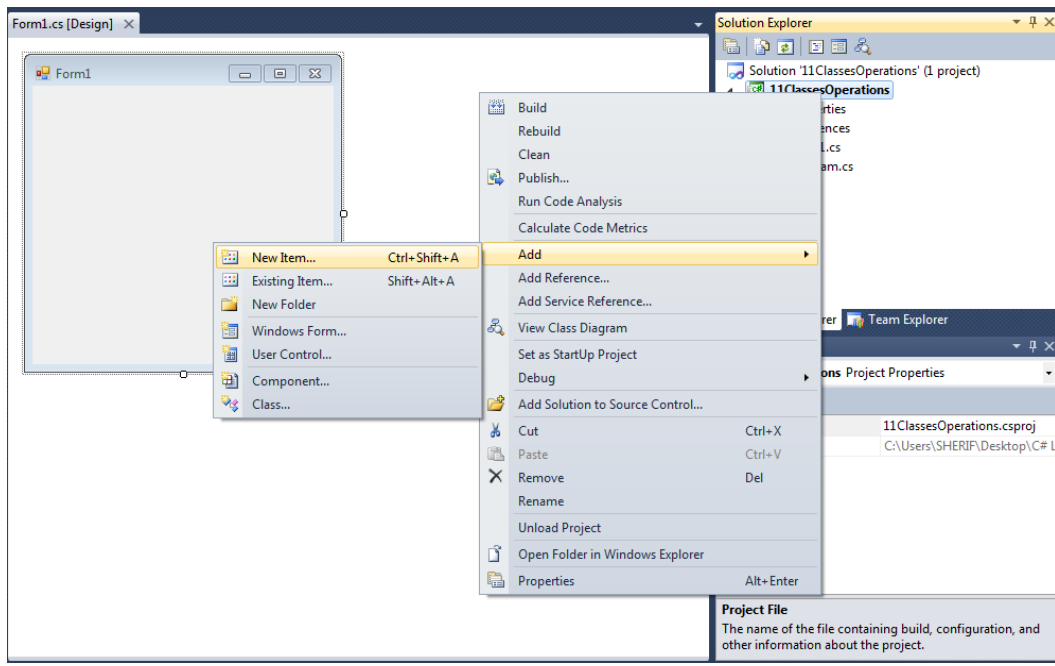
**لإستدعاء الفصيلة الخاصة بكل Namespace كالتالى :**

- 1- SunSchoolNameSpace. Student
- 2- MicrosoftSchoolNameSpace. Student

### **: Class**

**لتعريف الفصيلة أضف ملف جديد لمشروعك عن طريق الضغط بالماوس**

**Right-Click على اسم المشروع الخاص بك وإضافة New Item وحدد اسم الفصيلة وليكن Student كما بالشكل التالى :**



```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace _11ClassesOperations
{
    class Stubent
    {
```

```
}  
}
```

## : Object

يتم تعريفه بالصيغة التالية :

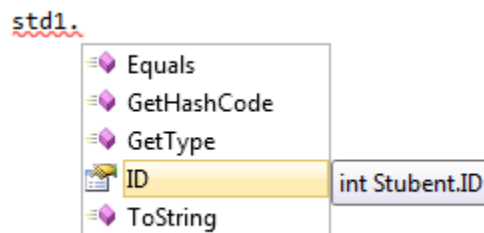
```
Student stu1 = new Student();
```

ويمكنه الوصول والتعامل مع خصائص الفصيله بعد أن قمنا بتعريفه .

## :Properties

تتم تعريفها بالصيغة التالية :

```
private int sid;  
  
public int ID  
{  
    set  
    {  
        sid = value;  
    }  
    get  
    {  
        return sid;  
    }  
}
```



وذلك للوصول للخصائص عن طريق اسم الهدف .



## :Constructor

يتم من خلاله إعطاء قيم ابتدائية للمتغيرات المستخدمة في الفصيلة ويأخذ نفس اسم الفصيلة ويتم تعريفه كالتالى :

```
public Student()  
{  
    sid = 0;  
    sname = "Name";  
    smajor = "Major";  
}
```

كما يمكن إعطاء قيم للمتغيرات أثناء التعريف كالتالى :

```
public Student(int id, string name, string  
major)  
{  
    sid = id;  
    sname = name;  
    smajor = major;  
}
```

## الشرط الثالث : Inheritance

هو أن ترث الفصيلة الابن Child Class خصائص ودوال الفصيلة الأب

Parent Class دون أن ترث الفصيلة الأب شيء من الفصيلة الابن أى أنها علاقة بين فصيليتين أحدهما يرث خصائص وصفات الفصيلة الأخرى دون أن يرث الأول منها شيء وصيغتها كالتالى :

```
public class ChildClass : ParentClass  
{  
    // محتوى class  
}
```

## الشرط الرابع : Polymorphism(Overloading - Overriding) :

### : Overloading

هو تعريف الدوال بنفس الاسم والنوع ولكن مع اختلاف عدد المتغيرات التي تأخذها الدالة .

مثال :

```
public double Average(int mark1, int mark2)
{
    return (mark1 + mark2) / 2;
}
```

```
public double Average(int mark1, int mark2, int mark3)
{
    return (mark1 + mark2 + mark3) / 3;
}
```

### : Overriding

وهو عبارة عن اعادة تعريف Method موجود داخل ال parent class بال child class بحيث يحمل نفس الاسم والنوع وعدد ونوع المتغيرات لكن بكود داخلي خاص بال child class. وهذا يعني أنه يتم استدعاء ال Method المعروف داخل ال child class فقط عند تعريف object من نوعه، وأيضا يتم استدعاء ال Method الذي يحمل نفس الاسم في ال parent class فقط عند تعريف object من نوعه، وكل منهم ينفذ الكود الخاص به. بالإضافة إلى أنه عند استدعاء ال Method سواء من student object أو parson object لن يظهر لنا إلا خيار واحد فقط ، وهو ال Method الموجود داخل هذه ال class .

مثال :

يتم تعريف الدالة في Parent Class مسبوقة بكلمة Virtual أى أن هذه الدالة حدث لها توارث داخل Child Class ولكن بكلمة override وهى نفس الدالة مع اختلاف محتوى الدالة داخل Child Class .

وصيغتها في Parent Class كالتالى :

```
public virtual string search(int id)
{
    return "person";
}
```

وصيغتها في Child Class كالتالى :

```
public override string search(int id)
{
    return "student";
}
```

## برنامج (12)

أنشئ مشروع جديد من قائمة File -> New -> Project من فيجوال ستوديو

واختر نوع المشروع Windows Forms Application باسم

ClassesOperations العمليات على الفصائل

ضع على الفورم التي تظهر لك الأدوات التالية وأعطى لها الخصائص الآتية:

### 1- Button:

Name: btnProperties

Text: Properties

Font Name: Times New Roman, 14pt, style=Bold

Size: 14

Bold: True

### 2- Button:

Name: btnConstructor

Text: Constructor

Font Name: Times New Roman, 14pt, style=Bold

Size: 14

Bold: True

### 3- Button:

Name: btnInheritance

Text: Inheritance

Font Name: Times New Roman, 14pt, style=Bold

**Size: 14**

**Bold: True**

#### **4- Button:**

**Name: btnOverloading**

**Text:Overloading**

**Font Name: Times New Roman, 14pt, style=Bold**

**Size: 14**

**Bold: True**

#### **5- Button:**

**Name: btnOverride**

**Text:Override**

**Font Name: Times New Roman, 14pt, style=Bold**

**Size: 14**

**Bold: True**

#### **6- Label:**

**Name: label1**

**Font Name: Times New Roman, 14pt, style=Bold**

**Size: 14**

**Bold: True**

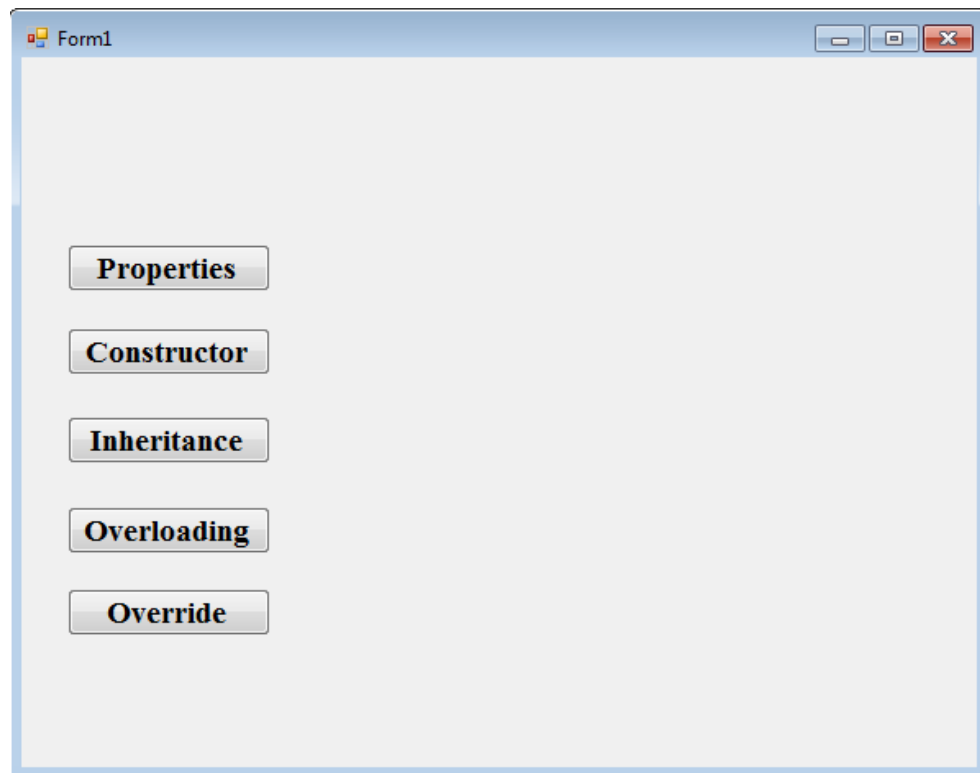
#### **7- Label:**

**Name: label2**

**Font Name: Times New Roman, 14pt, style=Bold**

Size: 14

Bold: True



أولا أضف إلى مشروعك الفصيلتين التاليتين :

### الفصيلة الأب Parent Class

```
public class Person
{
    // Properties For Inheritance
    int pssn;
    string address;

    public int SSN
    {
        get
        {
            return pssn;
        }
        set
        {
            pssn = value;
        }
    }
}
```

```

public string Address
{
    get
    {
        return paddress;
    }
    set
    {
        paddress = value;
    }
}

```

// Overriding Method

```

public virtual string search(int id)
{
    return "person";
}
}

```

## Child Class الفصيلة الابن

```

public class Student : Person
{
    private int sid;
    private string sname;
    private string smajor;

    // Properties

    public int ID
    {
        set
        {
            sid = value;
        }
        get
        {
            return sid;
        }
    }

    public string Name
    {
        set
        {
            sname = value;
        }
        get

```

```

        {
return sname;
        }
    }

public string Major
    {
set
        {
            smajor = value;
        }
get
        {
return smajor;
        }
    }

// Constructors
public Student()
    {
        sid = 20;
        sname = "Ali";
        smajor = "Pharmacy";
    }

public Student(int id, string name, string major)
    {
        sid = id;
        sname = name;
        smajor = major;
    }

// Overloading
public double Average(int mark1, int mark2)
    {
return (mark1 + mark2) / 2;
    }

public double Average(int mark1, int mark2, int mark3)
    {
return (mark1 + mark2 + mark3) / 3;
    }

```



```

    }

    // Overriding Method
    public override string search(int id)
    {
        return "Student";
    }
}

```

بالضغط مرتين بالماوس على Button Double ينشئ لنا

الحدث btnProperties\_Click ()

قم بكتابة الكود التالي بداخله كالتالي :

```

private void btnProperties_Click(object sender, EventArgs e)
{
    label2.Text = String.Empty;

    Student std1 = new Student();

    // للتعامل مع الخصائص Properties
    std1.ID = 111;
    std1.Name = "Mohamed";
    std1.Major = "Computer Science";

    label1.Text = "ID=" + std1.ID.ToString() + "\n" + "Name=" +
    std1.Name.ToString() + "\n" + "Major=" + std1.Major.ToString();
}

```

بالضغط مرتين بالماوس على Button Double ينشئ لنا

الحدث btnConstructor\_Click ()

قم بكتابة الكود التالي بداخله كالتالي :

```
private void btnConstructor_Click(object sender, EventArgs e)
{
    // للتعامل مع Constructor
    Student std2 = new Student();
    Student std3 = new Student(10, "Ahmed", "Engineering");

    label1.Text = "ID=" + std2.ID.ToString() + "\n" + "Name=" +
std2.Name.ToString() + "\n" + "Major=" + std2.Major.ToString();

    label2.Text = "ID=" + std3.ID.ToString() + "\n" + "Name=" +
std3.Name.ToString() + "\n" + "Major=" + std3.Major.ToString();
}
```

بالضغط مرتين بالماوس على Button Double ينشئ لنا

الحدث btnInheritance\_Click ()

قم بكتابة الكود التالي بداخله كالتالي :

```
private void btnInheritance_Click(object sender, EventArgs e)
{
    Person per1 = new Person();

    per1.SSN = 100;
    per1.Address = "Cairo";

    label1.Text = "SSN=" + per1.SSN.ToString() + "\n" + "Address=" +
per1.Address.ToString();

    Student std1 = new Student();

    std1.SSN = per1.SSN;
    std1.Address = per1.Address;

    label2.Text = "ID=" + std1.ID.ToString() + "\n" + "Name=" +
std1.Name.ToString() + "\n" + "Major=" + std1.Major.ToString() + "\n" + "SSN=" +
std1.SSN.ToString() + "\n" + "Address=" + std1.Address.ToString() + "\n";
}
```

```
}
```

بالضغط مرتين بالماوس على Button Double ينشئ لنا

الحدث btnOverloading\_Click ()

قم بكتابة الكود التالي بداخله كالتالي :

```
private void btnOverloading_Click(object sender, EventArgs e)
{
    // Method Overloading للتعامل مع

    Student std4 = new Student();
    double Avg1 = std4.Average(20, 20);
    double Avg2 = std4.Average(30, 30 ,30);

    label1.Text = "Average1=" + Avg1.ToString();
    label2.Text = "Average2=" + Avg2.ToString();
}
```

بالضغط مرتين بالماوس على Button Double ينشئ لنا

الحدث btnOverride\_Click ()

قم بكتابة الكود التالي بداخله كالتالي :

```
private void btnOverride_Click(object sender, EventArgs e)
{
    Student std1 = new Student();
    string std1String = std1.search(10);

    label1.Text = "Child Search =" + std1String;

    Person per1 = new Person();
    string per1String = per1.search(10);

    label2.Text = "Parent Search =" + per1String;
}
```

## (12) برمجة تطبيقات الويندوز

لبرمجة تطبيقات الويندوز بلغة C# لابد من التعامل مع بيئة التطوير Visual Studio باستخدام الأدوات Controls اللازمة لصنع برنامجك أو مشروعك الذى تقوم ببرمجته لذا فيتم هنا استخدام الأدوات الهامة لكل مبرمج يتطلع إلى الاحترافية فى برمجة الواجهات من خلال برامج تفصيلية لكل أداة أو عدة أدوات كما تتطلبه البرامج .

### الأدوات Controls :

#### 1 أداة الزر Button

وهى أداة يقوم بتنفيذ حدث معين أثناء الضغط عليها بالماوس أو تمرير الماوس عليها وهى ترتبط بأحداث الماوس مثل Click – DoubleClick وغيرها من الأحداث التى يتم تنشيطها عند الضغط على الزر أو استخدام أى حدث له ولها خواص مثل خاصية Text و Visible وغيرها من الخواص والتى يمكنك التعرف عليها من قائمة Properties فى Visual Studio .

### برنامج (13) :

أنشئ مشروع جديد من قائمة File -> New -> Project من فيجوال ستوديو واختر نوع المشروع Windows Forms Application باسم

PrintMessage

ضع على الفورم التى تظهر لك الأدوات التالية وأعطى لها الخصائص الآتية:

## 1- Button:

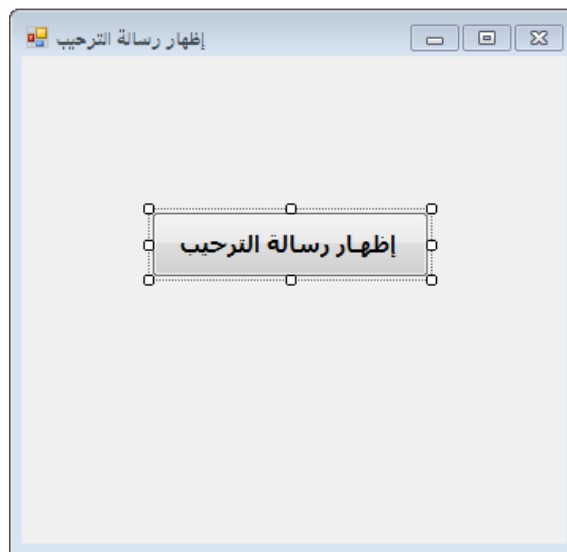
Name: btnMessage

Text: Properties

Font Name: Times New Roman, 14pt, style=Bold

Size: 14

Bold: True



بالضغط مرتين بالماوس على Button Double ينشئ لنا

الحدث btnMessage\_Click ()

قم بكتابة الكود التالي بداخله كالتالي :

```
private void btnMessage_Click(object sender, EventArgs e)
{
    lblMessage.Text = " مرحبا بكم في تعليم تطبيقات الوندوز بلغة C# ";
}
```

## برنامج (14) :

أنشئ مشروع جديد من قائمة File -> New -> Project من فيجوال ستوديو

واختار نوع المشروع Windows Forms Application باسم

### ButtonProperties

ضع على الفورم التي تظهر لك الأدوات التالية وأعطى لها الخصائص الآتية:

#### 1- Button:

Name: btnChangeColor

Text: تغيير لون خلفية الشاشة

Font Name: Times New Roman, 14pt, style=Bold

Size: 14

Bold: True

#### 2- Button:

Name: btnFont

Text: Change Form Font

Font Name: Times New Roman, 14pt, style=Bold

Size: 14

Bold: True

#### 3- Button:

Name: btnMessage

Text: إظهار رسالة

Font Name: Times New Roman, 14pt, style=Bold

Size: 14

Bold: True

#### 4- Button:

Name: btnExit

Text:خروج

Font Name: Times New Roman, 14pt, style=Bold

Size: 14

Bold: True



بالضغط مرتين بالماوس على Button Double ينشئ لنا

الحدث `btnChangeColor_Click ()`

قم بكتابة الكود التالي بداخله كالتالي :

```
privatevoid btnChangeColor_Click(object sender, EventArgs e)
{
    BackColor = Color.Green;
}
```

بالضغط مرتين بالماوس على Button Double ينشئ لنا

الحدث btnFont\_Click ()

قم بكتابة الكود التالي بداخله كالتالي :

```
privatevoid btnFont_Click(object sender, EventArgs e)
{
    this.Font = newFont("CASTELLAR", 16);
}
```

بالضغط مرتين بالماوس على Button Double ينشئ لنا

الحدث btnMessage\_Click ()

قم بكتابة الكود التالي بداخله كالتالي :

```
privatevoid btnMessage_Click(object sender, EventArgs e)
{
    MessageBox.Show("مرحباً بكم");
}
```

بالضغط مرتين بالماوس على Button Double ينشئ لنا

الحدث btnExit\_Click ()

قم بكتابة الكود التالي بداخله كالتالي :

```
privatevoid btnExit_Click(object sender, EventArgs e)
{
    Close();
}
```



## 2 صندوق النص TextBox

وهي عبارة عن أداة لإدخال النصوص والتي يتم من خلالها قراءة النص الذي يدخله المستخدم من الخاصية Text و TextAlign وغيرها من الخواص والتي يمكنك التعرف عليها من قائمة Properties في Visual Studio .

### برنامج (15) :

أنشئ مشروع جديد من قائمة File -> New -> Project من فيجوال ستوديو

واختر نوع المشروع Windows Forms Application باسم TextBox

ضع على الفورم التي تظهر لك الأدوات التالية وأعطى لها الخصائص الآتية:

#### 1- Button:

Name: btnChangeColor

Text: طباعة محتويات صندوق النص

Font Name: Times New Roman, 14pt, style=Bold

Size: 14

Bold: True

#### 2- TextBox:

Name: txtMessage

Font Name: Times New Roman, 14pt, style=Bold

Size: 14

Bold: True

RightToLeft: Yes



بالضغط مرتين بالماوس على Button Double ينشئ لنا

الحدث `btnMessage_Click ()`

قم بكتابة الكود التالي بداخله كالتالي :

```
private void btnPrintMessage_Click(object sender, EventArgs e)
{
    lblMessage.Text = txtMessage.Text;
}
```

### 3 صندوق النص المتعدد RichTextBox

وهي عبارة عن أداة لإدخال النصوص والتي يتم من خلالها قراءة النص الذي يدخله المستخدم وهي مثل أداة TextBox إلا أنها تحتوى على خصائص غير موجودة في أداة TextBox كونها تحتوى على نص متعدد الألوان والخطوط والأحجام ولها بعض الخواص مثل Text و TextAlign غيرها من الخواص والتي يمكنك التعرف عليها من قائمة Properties في Visual Studio .

#### برنامج (16) :

أنشئ مشروع جديد من قائمة File -> New -> Project من فيجوال ستوديو واختر نوع المشروع Windows Forms Application باسم RichTextBox ضع على الفورم التي تظهر لك الأدوات التالية وأعطى لها الخصائص الآتية:

#### 1- Button:

Name: btnRed

Text: لون الخط أحمر

Font Name: Times New Roman, 14pt, style=Bold

Size: 14

Bold: True

#### 2- Button:

Name: btnGreen

Text: لون الخط أخضر

Font Name: Times New Roman, 14pt, style=Bold

Size: 14

**Bold: True**

### **3- Button:**

**Name: btnBlue**

**Text:**لونا الخط أزرق

**Font Name: Times New Roman, 14pt, style=Bold**

**Size: 14**

**Bold: True**

### **4- Label:**

**Name: txtMessage**

**Text :** أكتب النص الذي تريده هنا

**Font Name: Times New Roman, 14pt, style=Bold**

**Size: 14**

**Bold: True**

### **5- TextBox:**

**Name: textbox1**

**Font Name: Times New Roman, 14pt, style=Bold**

**Size: 14**

**Bold: True**

**RightToLeft: Yes**

### **6- RichTextBox:**

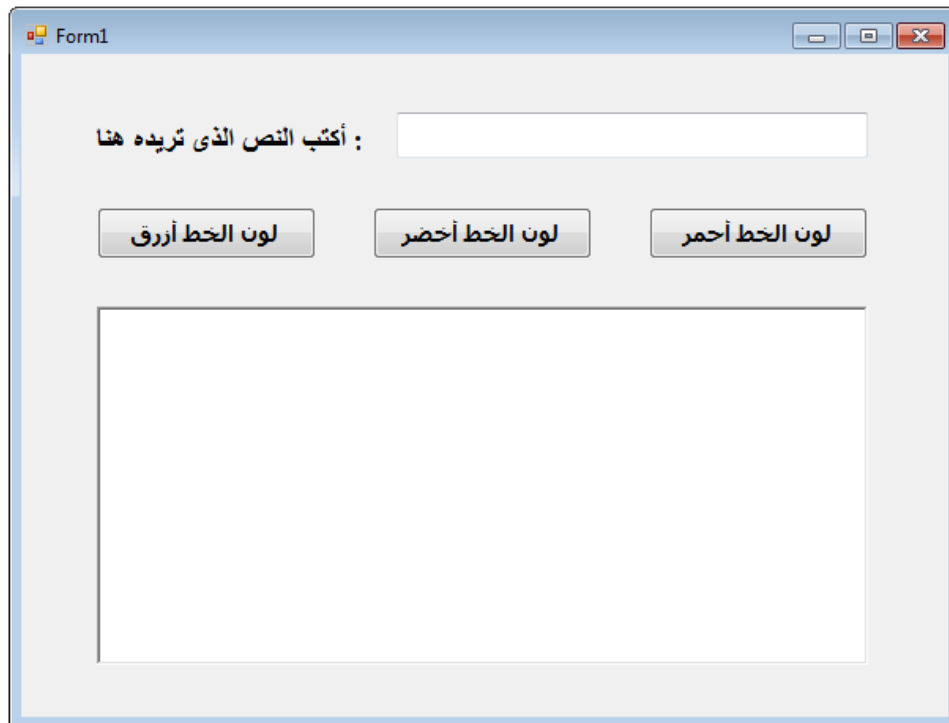
**Name: richTextBox1**

**Font Name: Times New Roman, 14pt, style=Bold**

**Size: 14**

**Bold: True**

**RightToLeft: Yes**



بالضغط مرتين بالماوس على Button btnRed ينشئ لنا

الحدث btnRed\_Click ()

قم بكتابة الكود التالي بداخله كالتالي :

```
private void btnRed_Click(object sender, EventArgs e)
{
    richTextBox1.SelectionFont = new Font("Timew New Roman", 40, FontStyle.Bold);
    richTextBox1.SelectionColor = Color.Red;
}
```

بالضغط مرتين بالماوس على Button btnGreen ينشئ لنا

الحدث btnGreen\_Click ()

قم بكتابة الكود التالي بداخله كالتالى :

```
privatevoid btnGreen_Click(object sender, EventArgs e)
{
    richTextBox1.SelectionFont = newFont("Timew New Roman", 40, FontStyle.Bold);
    richTextBox1.SelectionColor = Color.Green;
}
```

بالضغط مرتين بالماوس على Button btnBlue ينشئ لنا

الحدث btnBlue\_Click ()

قم بكتابة الكود التالي بداخله كالتالى :

```
privatevoid btnBlue_Click(object sender, EventArgs e)
{
    richTextBox1.SelectionFont = newFont("Timew New Roman", 40, FontStyle.Bold);
    richTextBox1.SelectionColor = Color.Blue;
}
```

بالضغط مرتين بالماوس على TextBox textbox1 ينشئ لنا

الحدث txtMessage\_TextChanged()

قم بكتابة الكود التالي بداخله كالتالى :

```
privatevoid txtMessage_TextChanged(object sender, EventArgs e)
{
    this.richTextBox1.Text = this.textbox1.Text;
}
```

#### 4 -أدوات إظهار النص Label , LinkLabel

وهى عبارة عنأداة تلعب دور مثل الملصق على الأجهزة الكهربائية أو الأدوات المنزلية فى إظهار النص وهى أداة Label أما الأداة الأخرى LinkLabel فهى نفس الأداة الأولى ولكن تزيد عليها كونها عبارة عن رابط انترنت ولهما بعض الخواص مثل Text و TextAlign غيرها من الخواص والتي يمكنك التعرف عليها من قائمة Properties فى Visual Studio .

#### برنامج (17) :

أنشئ مشروع جديد من قائمة File -> New -> Project من فيجوال ستوديو واختر نوع المشروع Windows Forms Application باسم

#### LabelAndLinkLabel

ضع على الفورم التى تظهر لك الأدوات التالية وأعطى لها الخصائص الآتية:

##### 1- Label:

Name: txtMessage

Text : This is a Label

Font Name: Times New Roman, 14pt, style=Bold

Size: 14

Bold: True

##### 2- LinkLabel:

Name: txtMessage

Text : This is a Link Label

Font Name: Times New Roman, 14pt, style=Bold

Size: 14

Bold: True



بالضغط مرتين بالماوس على LinkLabel linkLabel1 ينشئ لنا

الحدث linkLabel1\_LinkClicked ()

قم بكتابة الكود التالي بداخله كالتالي :

```
private void linkLabel1_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
{
    System.Diagnostics.Process.Start("www.google.com");
}
```



## 5 صندوق الاختيار CheckBox

وهي عبارة عن أداة تسمح للمستخدم بالاختيار من متعدد وتتعامل مع الخصائص التالية : Checked – Enabled و غيرها من الخواص والتي يمكنك التعرف عليها من قائمة Properties في Visual Studio .

### برنامج (18) :

أنشئ مشروع جديد من قائمة File -> New -> Project من فيجوال ستوديو

واختار نوع المشروع Windows Forms Application باسم

CheckBox

ضع على الفورم التي تظهر لك الأدوات التالية وأعطى لها الخصائص الآتية:

#### 1- Button:

Name: btnBlue

Text: لقد قممت باختيار

Font Name: Times New Roman, 14pt, style=Bold

Size: 14

Bold: True

#### 2- Label:

Name: txtMessage

Text : ما هيلغة البرمجة المفضلة لديك ؟

Font Name: Times New Roman, 14pt, style=Bold

Size: 14

**Bold: True**

### 3- CheckBox:

**Name: cb\_cs**

**Text:CSharp.Net**

**Font Name: Times New Roman, 14pt, style=Bold**

**Size: 14**

**Bold: True**

### 4- CheckBox:

**Name: cb\_vb**

**Text:Visual Basic.Net**

**Font Name: Times New Roman, 14pt, style=Bold**

**Size: 14**

**Bold: True**



بالضغط مرتين بالماوس على Button button1 ينشئ لنا

## الحدث button1\_Click ()

قم بكتابة الكود التالي بداخله كالتالى :

```
private void button1_Click(object sender, EventArgs e)
{

    if (cb_vb.Checked == true && cb_cs.Checked == false)
    {
        MessageBox.Show("Your Favourite language is : Visual Basic.net");
    }
    elseif (cb_vb.Checked == false && cb_cs.Checked == true)
    {
        MessageBox.Show("Your Favourite language is : CSharp.net");
    }
    elseif (cb_vb.Checked == false && cb_cs.Checked == false)
    {
        MessageBox.Show("Please Choose Your language !");
    }
    else
    {
        MessageBox.Show("Your Favourite language is : Visual Basic.net and CSharp.Net");
    }

}
```

## 6 زر الاختيار RadioButton

وهى عبارة عن أداة تشترك مع أداة CheckBox تقريبا فى الخواص إلا أنها لا تتيح للمستخدم إمكانية تعدد الاختيارات وتتعامل مع الخصائص التالية :  
Checked – Enabled وغيرها من الخواص والتي يمكنك التعرف عليها من قائمة Properties فى Visual Studio .

### برنامج (19) :

أنشئ مشروع جديد من قائمة File -> New -> Project من فيجوال ستوديو

واختر نوع المشروع Windows Forms Application باسم

**RadioButton**

ضع على الفورم التي تظهر لك الأدوات التالية وأعطى لها الخصائص الآتية:

### 1- Label:

Name: label1

Text:اختيار النوع:

Font Name: Times New Roman, 14pt, style=Bold

Size: 14

Bold: True

### 2- Button:

Name: button1

Text:اختيار كهو:

Font Name: Times New Roman, 14pt, style=Bold

Size: 14

Bold: True

### 3- RadioButton:

Name: radioButton1

Text:ذكر:

Font Name: Times New Roman, 14pt, style=Bold

Size: 14

Bold: True

### 4- RadioButton:

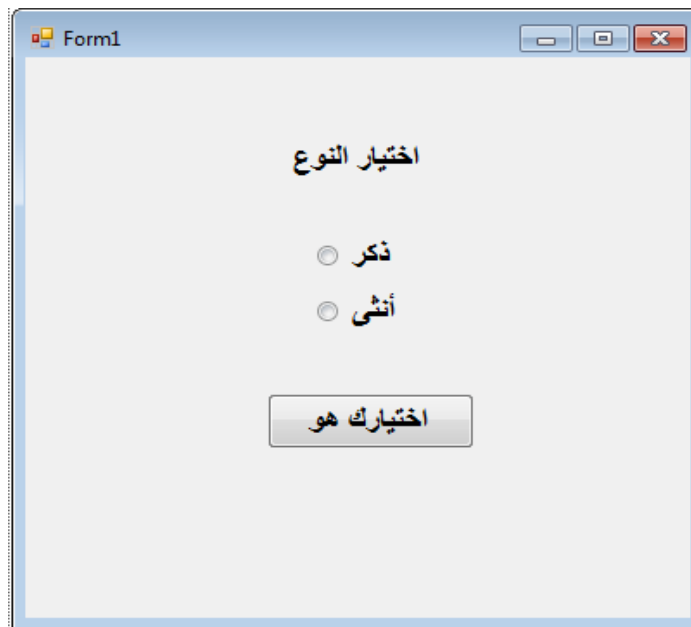
Name: radioButton2

**Text:** أنثى

**Font Name:** Times New Roman, 14pt, style=Bold

**Size:** 14

**Bold:** True



بالضغط مرتين بالماوس على Button button1 ينشئ لنا

الحدث button1\_Click ()

قم بكتابة الكود التالي بداخله كالتالي :

```
private void button1_Click(object sender, EventArgs e)
{
    if (radioButton1.Checked == true)
    {
        MessageBox.Show(" ذكر ");
    }
    else
    {
        MessageBox.Show(" أنثى ");
    }
}
```

## 7- علبة القائمة ListBox

تقوم هذه الأداة بعرض البيانات على شكل قائمة يمكن للمستخدم من خلالها أن يختار مايشاء ويمكن تعبئتها يدويا بأى عناصر كما تشاء من الخاصة Items وغيرها من الخواص والتي يمكنك التعرف عليها من قائمة Properties فى Visual Studio .

### برنامج (20) :

أنشئ مشروع جديد من قائمة File -> New -> Project من فيجوال ستوديو واختر نوع المشروع Windows Forms Application باسم

### ListBox1

ضع على الفورم التى تظهر لك الأدوات التالية وأعطى لها الخصائص الآتية:

#### 1- Label:

Name: label1

Text: قائمة هواياتك المفضلة

Font Name: Times New Roman, 14pt, style=Bold

Size: 14

Bold: True

#### 2- ListBox:

Name: listBox1

Font Name: Times New Roman, 14pt, style=Bold

Size: 14

Bold: True

Right-To-Left: Yes



بالضغط مرتين بالماوس على الفورم Form1 ينشئ لنا

الحدث Form1\_Load ()

قم بكتابة الكود التالي بداخله كالتالي :

```
private void Form1_Load(object sender, EventArgs e)
{
    this.listBox1.Items.Add("قراءة القصص الدينية");
    this.listBox1.Items.Add("السباحة");
    this.listBox1.Items.Add("ركوب الخيل");
}
```

## برنامج (21) :

أنشئ مشروع جديد من قائمة File -> New -> Project من فيجوال ستوديو

واختار نوع المشروع Windows Forms Application باسم

**ListBox2**

ضع على الفورم التي تظهر لك الأدوات التالية وأعطى لها الخصائص الآتية:

### 1- Button:

Name: button1

Text: أضف إلى القائمة

Font Name: Times New Roman, 14pt, style=Bold

Size: 14

Bold: True

### 2- Button:

Name: button2

Text: احذف العنصر الذي تم اختياره

Font Name: Times New Roman, 14pt, style=Bold

Size: 14

Bold: True

### 3- Button:

Name: button3

Text: حذف جميع العناصر

Font Name: Times New Roman, 14pt, style=Bold



**Size: 14**

**Bold: True**

#### **4- Label:**

**Name: label1**

**Text: أكتب اسمك هنا :**

**Font Name: Times New Roman, 14pt, style=Bold**

**Size: 14**

**Bold: True**

**Right-To-Left: Yes**

#### **5- TextBox:**

**Name: textBox1**

**Font Name: Times New Roman, 14pt, style=Bold**

**Size: 14**

**Bold: True**

**Right-To-Left: Yes**

#### **6- ListBox:**

**Name: listBox1**

**Font Name: Times New Roman, 14pt, style=Bold**

**Size: 14**

**Bold: True**

**Right-To-Left: Yes**



بالضغط مرتين بالماوس على الفورم Button button1 ينشئ لنا

الحدث button1\_Click ()

قم بكتابة الكود التالي بداخله كالتالي :

```
private void button1_Click(object sender, EventArgs e)
{
    this.listBox1.Items.Add(textBox1.Text);
    this.textBox1.Text = String.Empty;
}
```

بالضغط مرتين بالماوس على الفورم Button button2 ينشئ لنا

الحدث Button2\_Click ()

قم بكتابة الكود التالي بداخله كالتالي :

```
private void button2_Click(object sender, EventArgs e)
{
    this.listBox1.Items.Remove(listBox1.SelectedItem);
    //this.listBox1.Items.RemoveAt(listBox1.SelectedIndex);
}
```

بالضغط مرتين بالماوس على الفورم Button button3 ينشئ لنا

الحدث Button3\_Click ()

قم بكتابة الكود التالي بداخله كالتالى :

```
private void button3_Click(object sender, EventArgs e)
{
    this.listBox1.Items.Clear();
}
```

## 8 صندوق الكومبوجو ComboBox

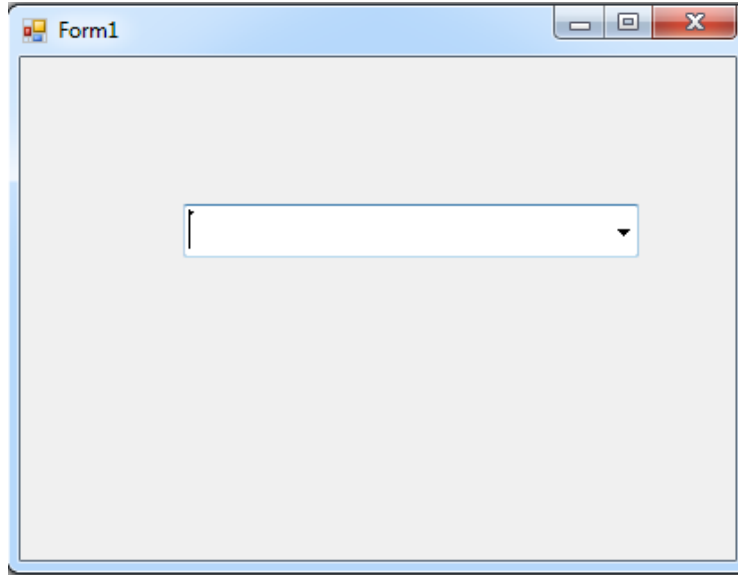
هذه الأداة تستعمل لتمكين المستخدم من اختيار بعض البيانات مثال عند التسجيل بموقع مث الياهو يطلب منك اختيار الدولة التى تنتمى إليها وخصائصها وأحداثها شبيهة بصندوق القوائم ListBox ويمكن تعبئتها يدويا بأى عناصر كما تشاء من الخاصة Items وغيرها من الخواص والتى يمكنك التعرف عليها من قائمة Properties فى Visual Studio .

### برنامج (22) :

أنشئ مشروع جديد من قائمة File -> New -> Project من فيجوال ستوديو

واختر نوع المشروع Windows Forms Application باسم

ComboBox



ضع على الفورم التى تظهر لك الأدوات التالية وأعطى لها الخصائص الآتية:

## 1- ComboBox:

Name: comboBox1

Font Name: Times New Roman, 14pt, style=Bold

Size: 14

Bold: True

بالضغط مرتين بالماوس على الفورم Form1 ينشئ لنا

الحدث Form1\_Load ()

قم بكتابة الكود التالى بداخله كالتالى :

```
privatevoid Form1_Load(object sender, EventArgs e)
{
    string[] Countries = { "مصر", "السعودية", "الجزائر", "العراق", "تونس", "ليبيا" };
    this.comboBox1.Items.AddRange(Countries);
}
```

بالضغط مرتين بالماوس على ComboBox comboBox1 ينشئ لنا

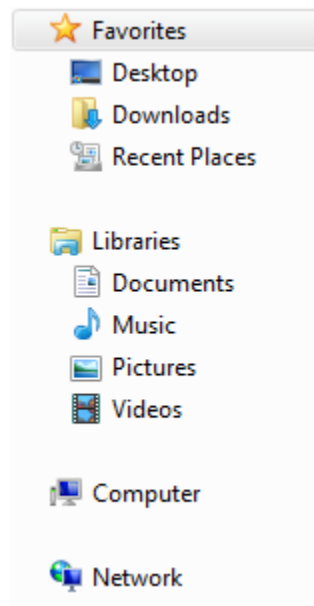
الحدث ( ) comboBox1\_SelectedIndexChanged

قم بكتابة الكود التالي بداخله كالتالى :

```
private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
{
    MessageBox.Show(" : لقد اخترت دولة " + comboBox1.Text);
}
```

## 9- القائمة الشجرية TreeView

سبق وأن شاهدت هذه الأداة مع متصفح الويندوز



وهي أداة تقوم بإظهار البيانات تنازليا على شكل بيانات شجرية متشعبة من الأصل إلى الفروع وتستعمل غالبا في تصفح وحدات الحاسوب Drives والمجلدات Folders والملفات Files ولها العديد من الخواص مثل Nodes و Right-To-Left وغيرها من الخواص والتي يمكنك التعرف عليها من قائمة Properties فى Visual Studio .

## برنامج (23) :

أنشئ مشروع جديد من قائمة **File -> New -> Project** من فيجوال ستوديو

واختار نوع المشروع **Windows Forms Application** باسم

**TreeView1**

ضع على الفورم التي تظهر لك الأدوات التالية وأعطى لها الخصائص الآتية:

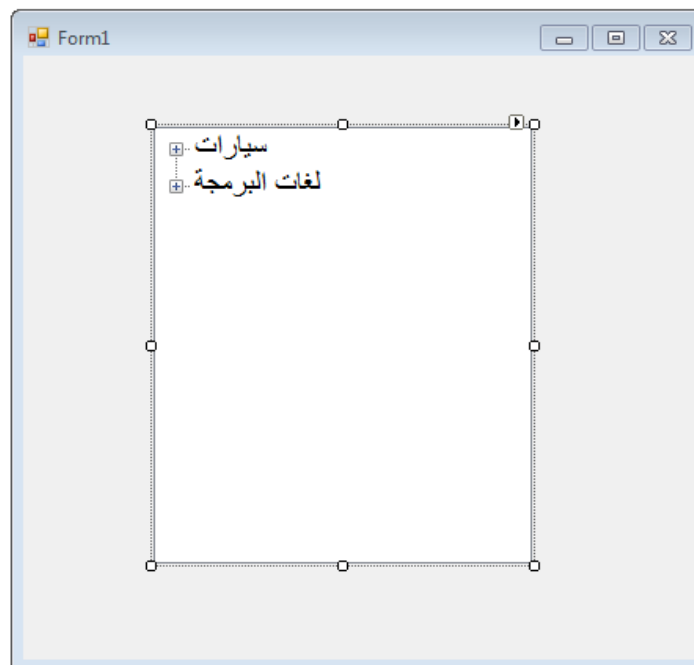
### 1- TreeView:

**Name:** treeView1

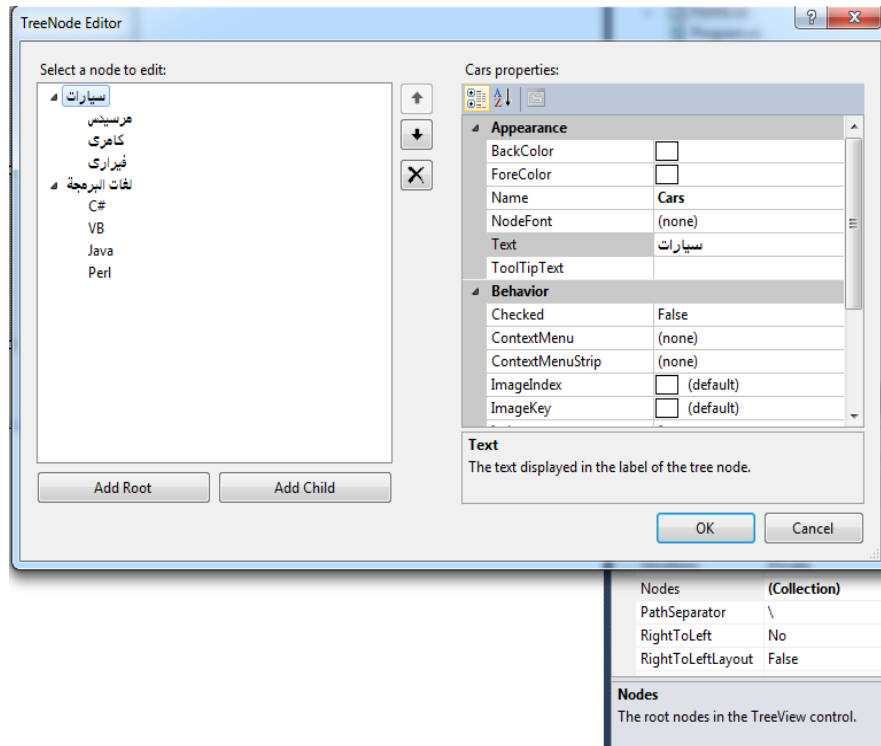
**Font Name:** Times New Roman, 14pt, style=Bold

**Size:** 14

**Bold:** True



أضف عناصر للقائمة الشجرية من الخاصية Nodes كالتالى :



**برنامج (24) :**

أنشئ مشروع جديد من قائمة **File -> New -> Project** من فيجوال ستوديو  
واختر نوع المشروع **Windows Forms Application** باسم

**TreeView2**

ضع على الفورم التى تظهر لك الأدوات التالية وأعطى لها الخصائص الآتية:

## 1- Label:

**Name:** label1

**Text:** انقر بالماوس مرتين لإختيار العنصر من الشجرة

**Font Name:** Times New Roman, 14pt, style=Bold

**Size:** 14

**Bold:** True

## 2- ImageList:

**Name:** imageList1

**Images:** (Inside Image Collection Editor put Image called Folder.jpg)

## 3- TreeView:

**Name:** treeView1

**ImageList:** imageList1

**ImageIndex:** 0

**Font Name:** Times New Roman, 14pt, style=Bold

**Size:** 14

**Bold:** True

**Right-To-Left:** Yes

**Right-To-Left-Layout:** True

**SelectedImageIndex:** 0





بالضغط مرتين بالماوس على الفورم Form1 ينشئ لنا

الحدث Form1\_Load ()

قم بكتابة الكود التالي بداخله كالتالي :

```
privatevoid Form1_Load(object sender, EventArgs e)
{

    TreeNode tNode;
    tNode = treeView1.Nodes.Add("Websites");

    treeView1.Nodes[0].Nodes.Add("Net-informations.com");
    treeView1.Nodes[0].Nodes[0].Nodes.Add("CLR");

    treeView1.Nodes[0].Nodes.Add("Vb.net-informations.com");
    treeView1.Nodes[0].Nodes[1].Nodes.Add("String Tutorial");
    treeView1.Nodes[0].Nodes[1].Nodes.Add("Excel Tutorial");

    treeView1.Nodes[0].Nodes.Add("Csharp.net-informations.com");
    treeView1.Nodes[0].Nodes[2].Nodes.Add("ADO.NET");
    treeView1.Nodes[0].Nodes[2].Nodes[0].Nodes.Add("Dataset");

}
```

بالضغط مرتين بالماوس على `TreeView treeView1` ينشئ لنا

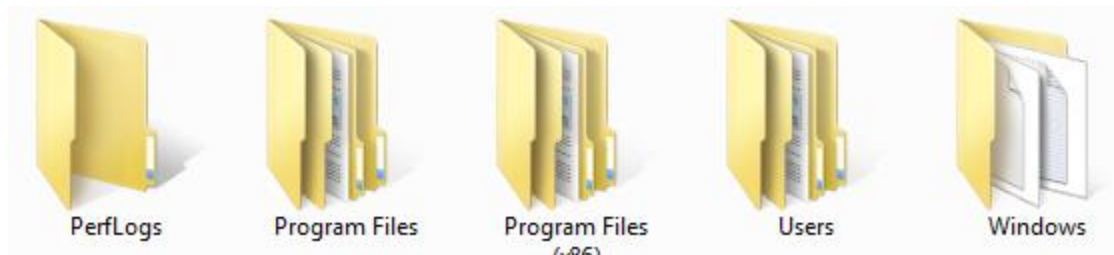
الحدث `treeView1_MouseDoubleClick()`

قم بكتابة الكود التالي بداخله كالتالي :

```
private void treeView1_MouseDoubleClick(object sender, MouseEventArgs e)
{
    TreeNode Node = treeView1.SelectedNode;
    MessageBox.Show(string.Format("لقد اخترت: {0}", Node.Text));
}
```

## 10 - قائمة العرض ListView

هذه الأداة شبيهة بصندوق القوائم `ListBox` إلا أنها أفضل منها حيث أنها تستطيع عرض البيانات بطريقة منسقة وجذابة كما تستطيع عرض الصور كما في متصفح الويندوز كما في متصفح الويندوز مصمم بهذه الأداة .



### برنامج (25) :

أنشئ مشروع جديد من قائمة `File -> New -> Project` من فيجوال ستوديو

واختار نوع المشروع `Windows Forms Application` باسم

`ListView1`

ضع على الفورم التي تظهر لك الأدوات التالية وأعطى لها الخصائص الآتية:

## 1- Label:

Name: label1

Text: استعراض المجلدات

Font Name: Times New Roman, 14pt, style=Bold

Size: 14

Bold: True

## 2- ComboBox:

Name: comboBox1

DropDownStyle: DropDownList

Items:

(Collection)

(LargeIcon

Details

SmallIcon

List

Title)

Font Name: Times New Roman, 14pt, style=Bold

Size: 14

Bold: True

## 3- ImageList:

Name: imageList1

Images: (Inside Image Collection Editor put Image called Folder.jpg)

## 4- ListView:

**Name:** listView1

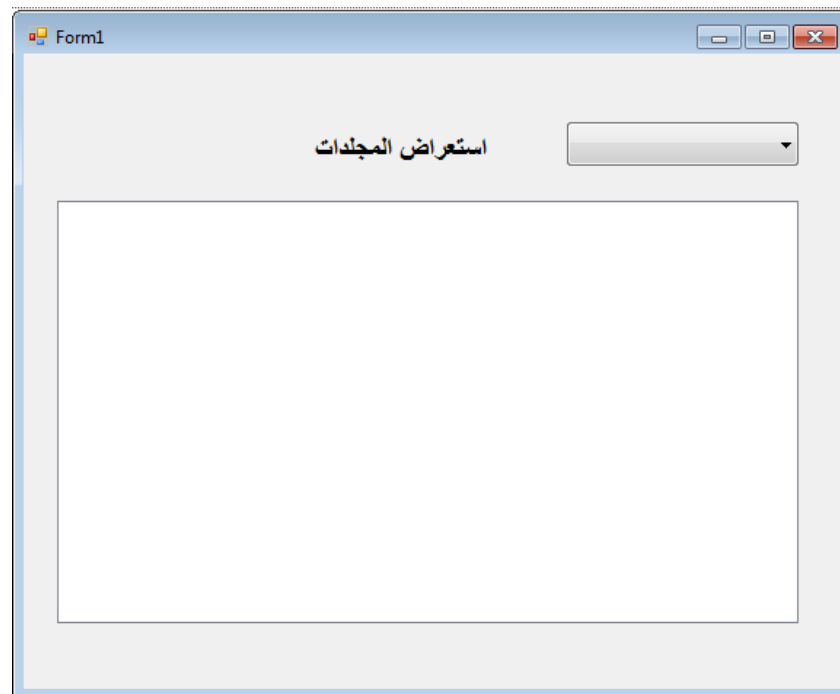
**LargeImageList:** imageList1

**SmallImageList:** imageList1

**Font Name:** Times New Roman, 14pt, style=Bold

**Size:** 14

**Bold:** True



## بكتابة هذا الكود داخل Constructor Form1() لإظهار أسماء وتفاصيل المجلدات داخل ListView

```
public Form1()
{
    InitializeComponent();

    listView1.Columns.Add("الإسم");
    listView1.Columns.Add("التفاصيل");

    listView1.Items.Add("المجلد 1", 0);
    listView1.Items.Add("المجلد 2", 0);
    listView1.Items.Add("المجلد 3", 0);
    listView1.Items.Add("المجلد 4", 0);
    listView1.Items.Add("المجلد 5", 0);
}
```

بالضغط مرتين بالماوس على ComboBoxcomboBox1 ينشئ لنا

الحدث comboBox1\_SelectedIndexChanged()

قم بكتابة الكود التالي بداخله كالتالي :

```
private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
{
    if (comboBox1.Text == "LargeIcon")
    {
        listView1.View = View.LargeIcon;
    }
    elseif (comboBox1.Text == "Details")
    {
        listView1.View = View.Details;
    }
    elseif (comboBox1.Text == "SmallIcon")
    {
        listView1.View = View.SmallIcon;
    }
    elseif (comboBox1.Text == "List")
    {
        listView1.View = View.List;
    }
    else
    {
        listView1.View = View.Tile;
    }
}
```

## برنامج (26) :

أنشئ مشروع جديد من قائمة **File -> New -> Project** من فيجوال ستوديو

واختار نوع المشروع **Windows Forms Application** باسم

**ListView2**

ضع على الفورم التي تظهر لك الأدوات التالية وأعطى لها الخصائص الآتية:

### **1- ImageList:**

**Name:** imageList1

**Images:** (Inside Image Collection Editor put Images called →

**Drive.png, Folder.png, File.png)**

### **2- ListView:**

**Name:** lvDrives

**LargeImageList:** imageList1

**SmallImageList:** imageList1

**Font Name:** Times New Roman, 14pt, style=Bold

**Size:** 14

**Bold:** True

### **3- ListView:**

**Name:** lvFiles

**LargeImageList:** imageList1

**SmallImageList:** imageList1

**Font Name:** Times New Roman, 14pt, style=Bold

**Size: 14**

**Bold: True**

#### **4- TreeView:**

**Name: tvFolders**

**ImageIndex: 0**

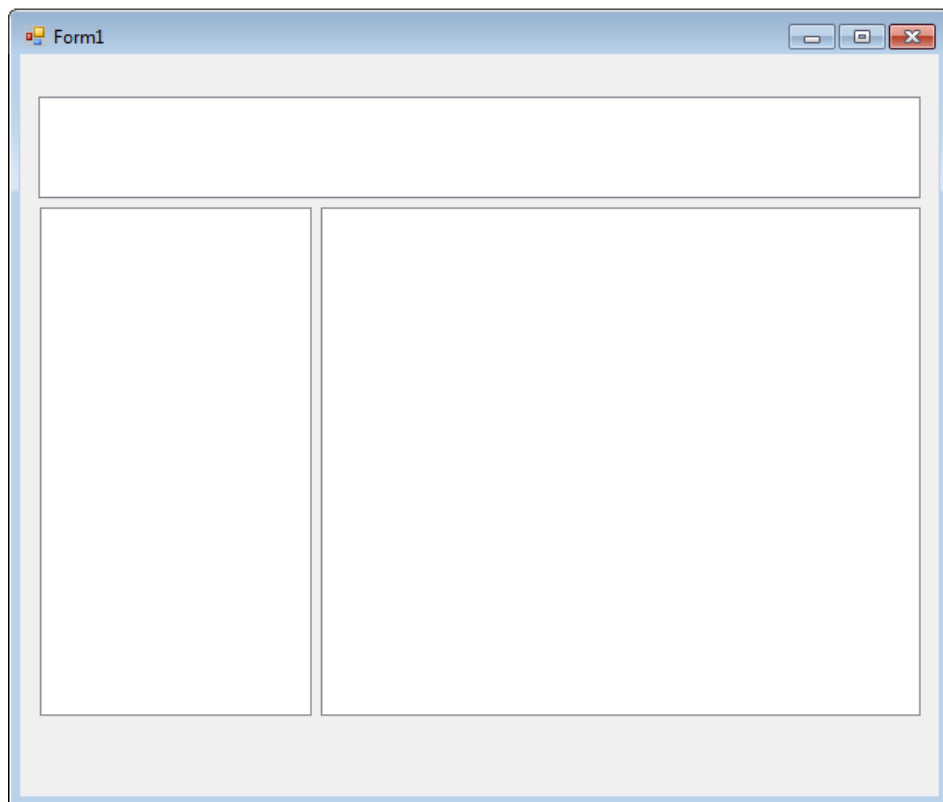
**ImageList: imageList1**

**SelectedImageIndex: 0**

**Font Name: Times New Roman, 14pt, style=Bold**

**Size: 14**

**Bold: True**



## قم بكتابة أكواد الدوال التالية للتعامل مع وحدات الحاسوب Drives و المجلدات Folders والملفات Files كالتالى :

```
string strDrive;

// First Method
void getDrives()
{
    try
    {
        foreach (string drive in Directory.GetLogicalDrives())
        {
            this.lvDrives.Items.Add(drive, 0);
        }
    }

    catch (Exception ex)
    {
        ex.Message.ToString();
        lblError.Text = "لم يتم الوصول إلى وحدات الحاسب";
    }
}

// Second Method
void getFolders()
{
    try
    {
        strDrive = lvDrives.FocusedItem.Text;
        DirectoryInfo Dir = new DirectoryInfo(strDrive);
        tvFolders.Nodes.Clear();
        foreach (DirectoryInfo Folder in Dir.GetDirectories())
        {
            tvFolders.Nodes.Add("", Folder.Name, 1, 1);
        }
    }

    catch (Exception ex)
    {
        ex.Message.ToString();
        lblError.Text = "لم يتم الوصول إلى مجلدات الحاسب";
    }
}

// Third Method
void getFiles(string strPath)
{
    try
    {

```



```

ListViewItem lvi;
DirectoryInfo Dir = newDirectoryInfo(strDrive + strPath);
lvFiles.Items.Clear();
foreach (FileInfo file in Dir.GetFiles())
{
    lvi = lvFiles.Items.Add(file.Name, 2);
    lvi.SubItems.Add(file.Length.ToString());
    lvi.SubItems.Add(file.LastAccessTime.ToString());
    lvi.SubItems.Add(file.Extension);
}

}

catch (Exception ex)
{
    ex.Message.ToString();
    lblError.Text = "لم يتم الوصول إلى الملف";
}

}

```

**قم بإستدعاء الدالة getDrives() داخل Constructor Form1() كالتالى :**

```

public Form1()
{
    InitializeComponent();

    getDrives();
}

```

**باختيار الحدث AfterSelect الخاص بالأداة TreeView tvFolders**

**قم بكتابة الكود التالى بداخله كالتالى :**

```

privatevoid tvFolders_AfterSelect(object sender, TreeViewEventArgs e)
{
    try
    {
        getFiles(e.Node.FullPath);
        TreeNode node;
        DirectoryInfo Dir = newDirectoryInfo(strDrive +
            e.Node.FullPath);
        foreach (DirectoryInfo folder in Dir.GetDirectories())
        {
            node = newTreeNode(folder.Name, 1, 1);
            e.Node.Nodes.Add(node);
        }
    }
    catch (Exception ex)
    {
    }
}

```

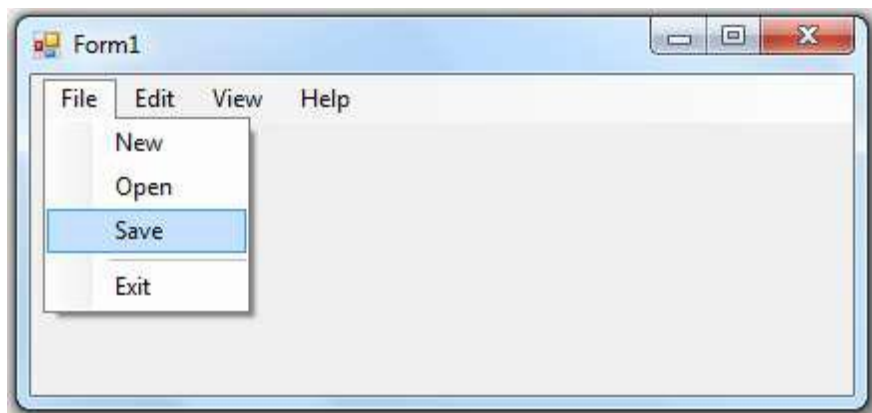
```

ex.Message.ToString();
lblError.Text = "لم يتم الوصول إلى المجلدات الحاسب";
}
}

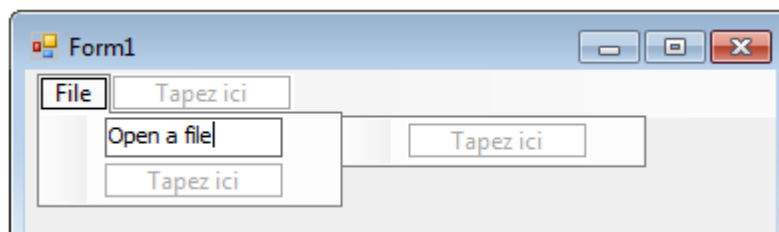
```

## 11 - أداة القائمة الرئيسية MenuStrip

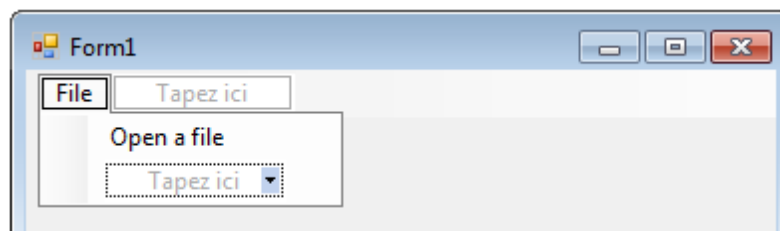
تعد هذه الأداة من من أهم الأدوات التي يحفل بها معظم البرامج وهذه صورة لها :



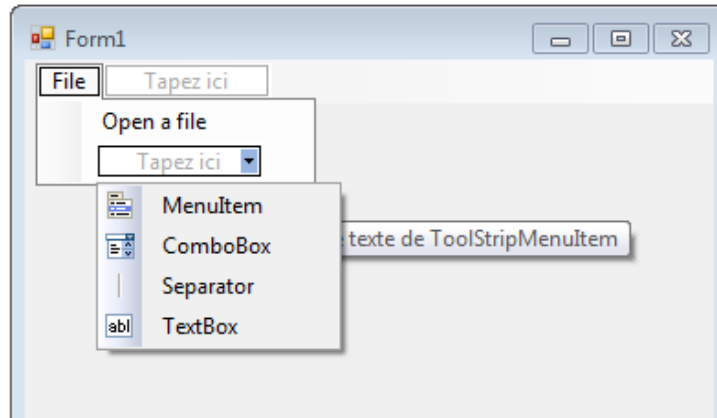
من علبة الأدوات ToolBox قم بجذب أداة القائمة MenuStrip إلى الفورم جرب أن تضيف بعض القوائم الرئيسية وبعض القوائم الفرعية لها



إذا مررت بال مؤشر فوق مكان كتابة اسم القائمة سيظهر لك سهم صغير موجه نحو الأسفل كما تظهر الصورة التالية :



إذا قمت بالضغط عليه ستظهر لك الإختيارات التالية :



### برنامج (27) :

أنشئ مشروع جديد من قائمة **File -> New -> Project** من فيجوال ستوديو

واختر نوع المشروع **Windows Forms Application** باسم

### **MenuStrip**

ضع على الفورم التي تظهر لك الأدوات التالية وأعطى لها الخصائص الآتية:

#### **1- MenuStrip:**

Name: menuStrip1

Text : menuStrip1

#### **2- fileToolStripMenuItem:**

Name: fileToolStripMenuItem

Text: &File

#### **3-openToolStripMenuItem**

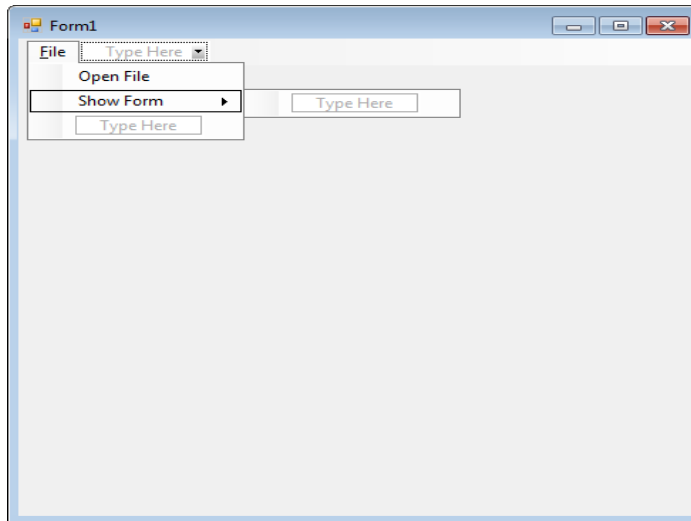
Name: openToolStripMenuItem

**Text: Open File**

#### **4-showFormToolStripMenuItem**

**Name: showFormToolStripMenuItem**

**Text: Show Form**



بالماسعلنزر بالضغط مرتين

**Button showFormToolStripMenuItem**

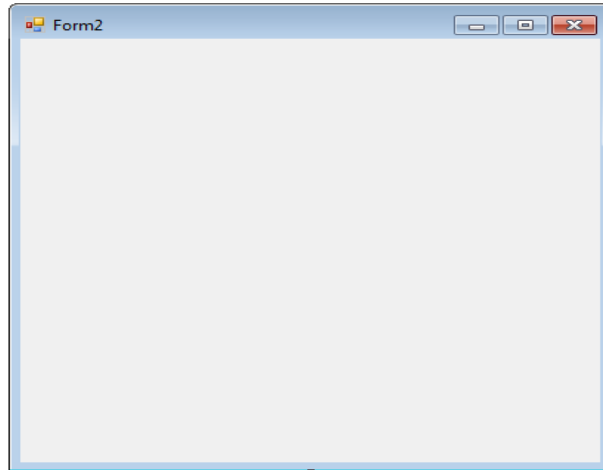
ينشئ لنا الحدث

**showFormToolStripMenuItem\_Click()**

قم بكتابة الكود التالي بداخله كالتالي :

```
private void showFormToolStripMenuItem_Click(object sender, EventArgs e)
{
    Form2 frm = new Form2();
    frm.Show();
}
```

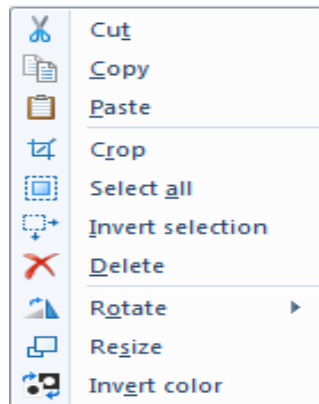
إنشاء فورم أخرى Form2 يتم مناداتها من خلال الزر الموجود بالفورمة السابقة



## 12 - أداة القائمة المنسدلة ContextMenuStrip

طبعا سبق لك وأن رأيت القائمة المنسدلة بالضغط على يمين الماوس

كالتالى :



## برنامج (28) :

أنشئ مشروع جديد من قائمة **File -> New -> Project** من فيجوال ستوديو

واختار نوع المشروع **Windows Forms Application** باسم

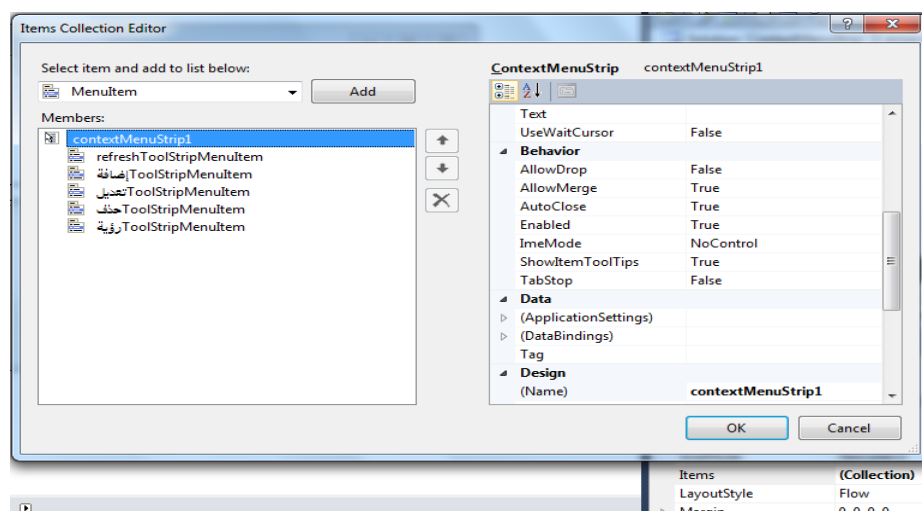
### **ContextMenuStrip**

ضع على الفورم التي تظهر لك الأدوات التالية وأعطى لها الخصائص الآتية:

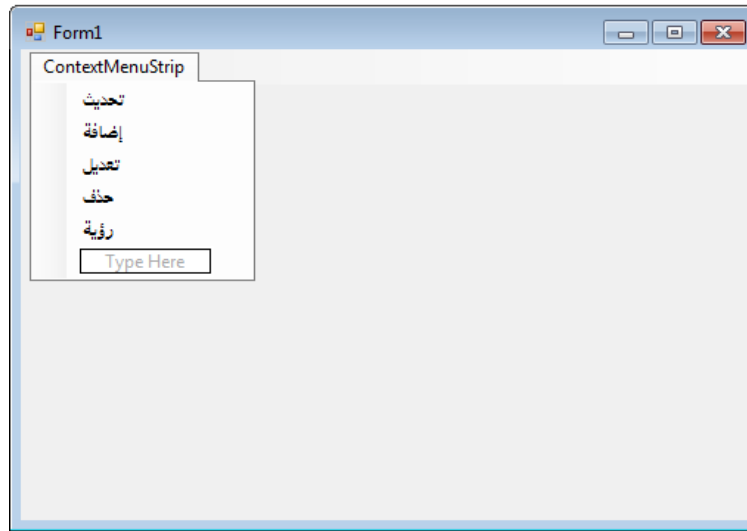
#### **1- ContextMenuStrip:**

**Name:** contextMenuStrip1

**Items:** Collection



عند التصميم تظهر الفورم فى التصميم بالشكل التالى :



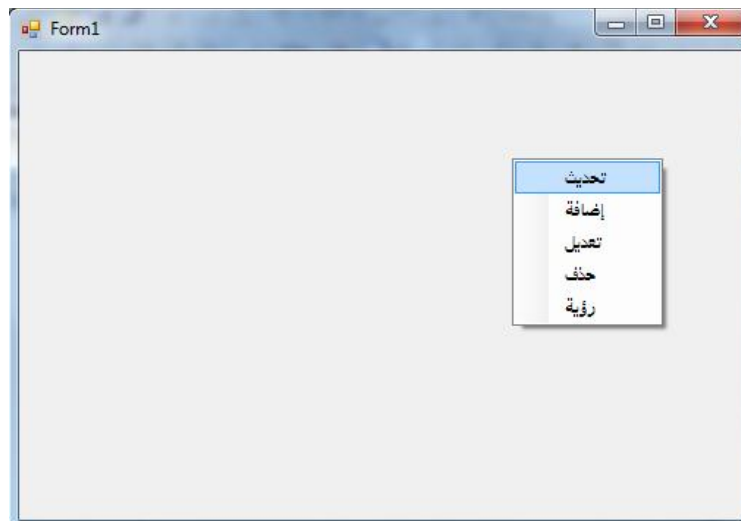
## 2-Form1

### ContextMenuStrip: contextMenuStrip1

Name: Form1

Text: Form1

عند تشغيل البرنامج تظهر الفورم بالشكل التالى :



## 13 - أداة شريط الأدوات ToolStrip

غالبا ما يكون الغرض من هذه الأداة هو عرض محتوى القوائم الفرعية بالصور وبشكل واضح يجعل المستخدم أكثر ارتياحا وهذه صورة الأداة .



### برنامج (29) :

أنشئ مشروع جديد من قائمة **File -> New -> Project** من فيجوال ستوديو  
واختر نوع المشروع **Windows Forms Application** باسم

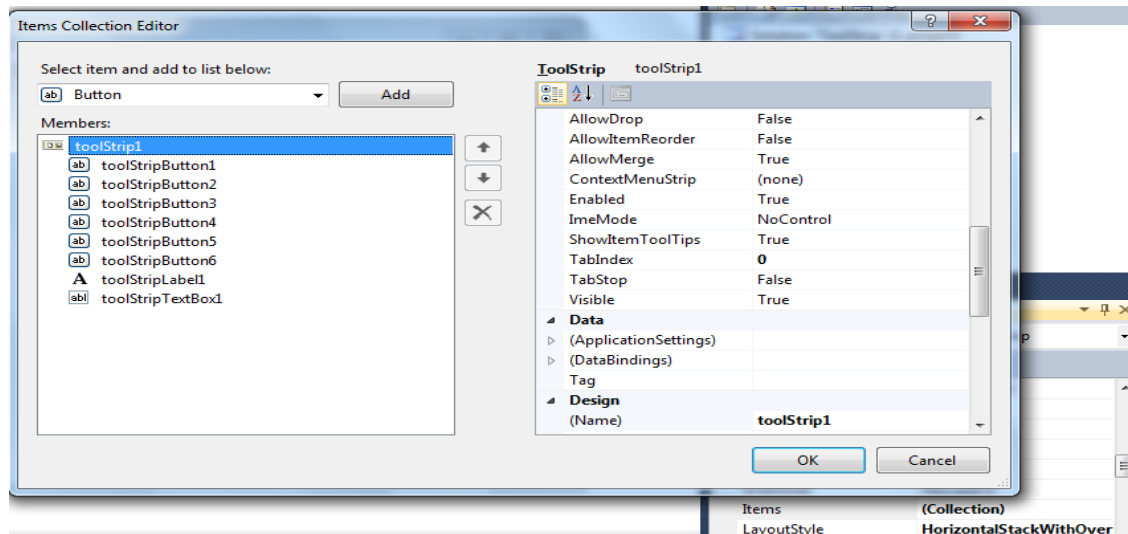
## ToolStrip

ضع على الفورم التي تظهر لك الأدوات التالية وأعطى لها الخصائص الآتية:

### 1- ToolStrip

Name: toolStrip1

Items: Collection





**LayoutStyle: HorizontalStackWithOverflow**

**Text: toolStrip1**

## **1- Label:**

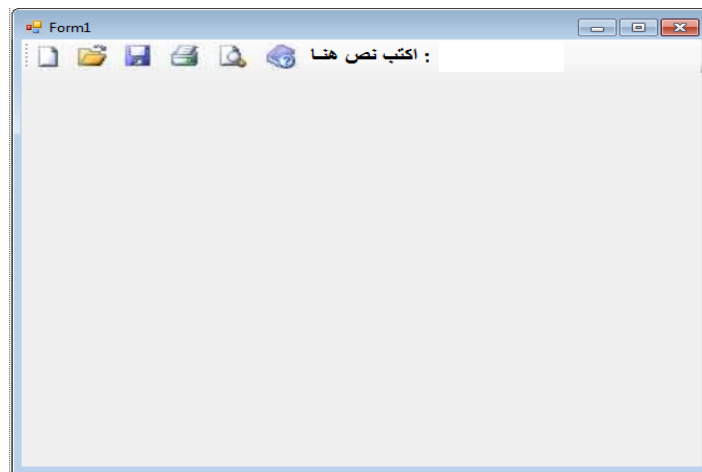
**Name: label1**

**Text:**

**Font Name: Times New Roman, 14pt, style=Bold**

**Size: 14**

**Bold: True**



بالضغط مرتين بالماوس على toolStripTextBox1 ينشئ لنا

**الحدث ( ) toolStripTextBox1\_TextChanged**

**قم بكتابة الكود التالي بداخله كالتالي :**

```
private void toolStripTextBox1_TextChanged(object sender, EventArgs e)
{
    this.label1.Text = toolStripTextBox1.Text;
}
```

نتناول في هذا الكتاب شرح ربط برامج السي شارب بقواعد البيانات الأكسس والإس كيو إل سرفر وحتى وإن كان أغلب الشرح هو بالإكسس ولكن ركزوا على هذه النقطة أن التغيير البسيط في الشرح مابين القاعدتين هو اختلاف مجال الأسماء واختلاف نص الاتصال و تستطيع استخدام الشرح المقدم في هذا الكتاب في لغة الاس كيو إل سرفر فما عليك بعد النقطتين السابقتين إلا أن تغير اشتقاق الكائنات التي تتبع الـ (OLDEB) مثلاً فئة (OleDbConnection) هي فئة الإتصال باستخدام قاعدة بيانات من نوع الأكسس والاختلاف انه عند استخدام قاعدة بيانات الإس كيو إل سرفر اننا نغير كلمة (OLDEB) بكلمة (SQL) يعني سيصبح هكذا (SqlConnection) وفي كل الفئات الذي تبدأ بكلمة (OLDEB) ما علينا سوى إستبدالها بكلمة (SQL) .

وإن ما دعاني لأكتب هذا الكتاب هو ما لاقيته من صعوبة في وجود مناهج بالعربية تقدم الشرح المبسط لهذه الإجراءات وأرجوا أن يستفيد من ما كتبت في هذا الكتاب أي شخص وأعذروني عن أي تقصير وأن كان هناك أية أسئلة أرجوا منكم أن تراسلوني على احد الإيميلات .

امتناني لكم

حمزة علي محمد عبده حزام البعسي

برمجة حاسوب - اليمن

سنبدأ أول درس من دروسنا البسيطة بدرس كيفية إنشاء قاعدة بيانات بسيطة في الأكسس .

نضغط على قائمة ابدأ ثم كافة البرامج ثم Microsoft Office ثم Microsoft Office Access وبعد أن تفتح لنا واجهة البرنامج نضغط على ( قاعدة بيانات فارغة ) كما في الصورة



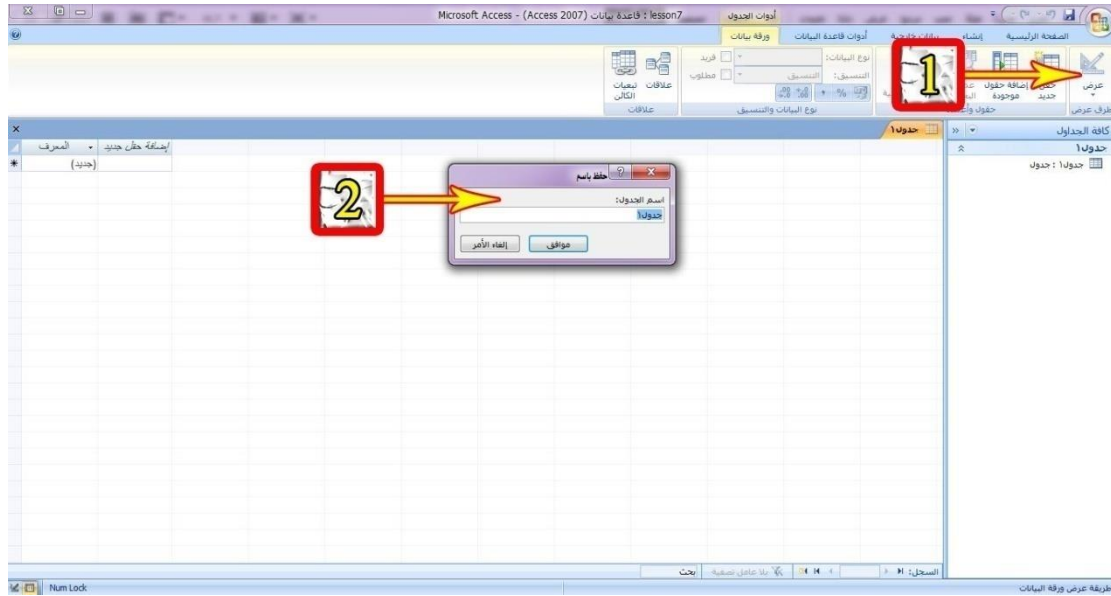
1 / قاعدة بيانات فارغة جديدة

2 / تسمية قاعدة البيانات بالاسم الذي تشاء

3 / مكان حفظ قاعدة البيانات التي سوف تنشئها اضغط على استعراض واختار المكان الذي تريده

وفي هذا المثال أنا استخدمت الاسم lesson7 ومن ثم الضغط على إنشاء ليتم إنشاء قاعدة البيانات الخاصة بك .

- تفتح لنا هذه الواجهة



- 1 / الضغط على (عرض) في أعلى يمين الشاشة لإظهار تصميم الجدول الجديد
- 2 / يطلب منا أن نقوم بتسمية الجدول الجديد اختر الاسم الذي تريده .

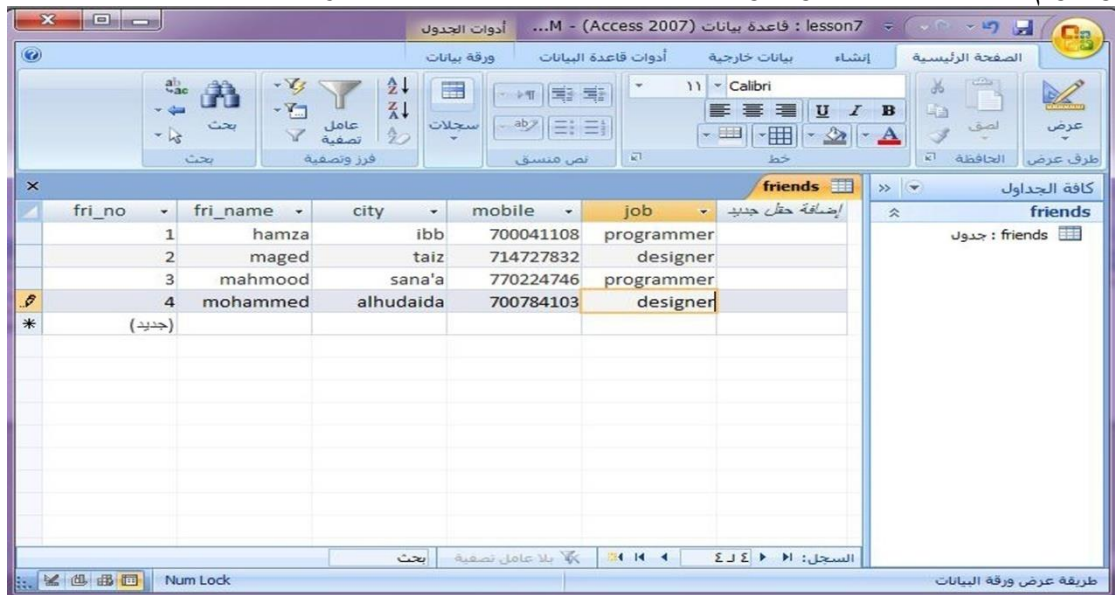
و أنا قد أسميت الجدول الذي سوف نقوم بالتطبيق عليه بالاسم friends  
بعد ذلك تفتح لنا صفحة التصميم فنقوم بتصميم جدولنا كما نريد وكما نحتاج من  
أعمدة وأنواعها البيانية



- 1 / قمنا بتعبئة أسماء الأعمدة في الجدول والنوع البياني لها حسب احتياجنا

- واختارنا رقم الصديق أو (fri\_no) ليكون مفتاح أساسي
- 2 / نعود لعرض الجدول في طريقة عرض ورقة البيانات بالضغط على ( عرض)  
سوف تظهر لنا هل تريد الحفظ ان لم نكن قد حفظنا ماقمنا به بنوافق بالضغط على نعم .  
ليظهر لنا الجدول باعمدته خالية من البيانات نقوم بتعبئتها بالبيانات

ونقوم بعدها بالحفظ ويكون شكلها قبل الحفظ هكذا تقريباً



نغلق برنامج الأكسس .

قبل ان ندخل في الربط وطريقته يجب ان نعرف بعض الأمور عن هذه التقنية التي ابتكرتها شركة مايكروسوفت وهي تقنية ado.net وهي التقنية التي نستخدمها في عملية الربط .

هناك ما يسمى بالأعمدة السبعة لتقنية ado.net من مايكروسوفت وهي كلاسات او فئات تقدمها لنا للتحكم بعملية ربط برامج الواجهات مع قواعد البيانات المختلفة وهي

1. فئة الاتصال (connection) .

وهي الفئة المخصصة للاتصال بقواعد البيانات فيحدد مسار قاعدة البيانات وما يتعلق بذلك من وسائط كالتشفير واسم المستخدم وغير ذلك .

2. فئة الأمر (command) .

وهي الفئة المخصصة لتنفيذ أمر ما من أوامر SQL المختلفة , أو أي إجراء مخزن في نظام إدارة قاعدة البيانات .

3. فئة قارئ البيانات (Data Reader) .

تستخدم للقراءة فقط من قاعدة البيانات في النمط المتصل , والبيانات يمكن قراءتها إلى الأمام ولا يمكن الرجوع إلى الخلف , وذلك بفئة الاستعلام وتعتبر ذات أداء عالي جداً .

4. فئة مهيي ( موصل ) البيانات (Data Adapter) .

تعمل هذه الفئة على تعبئة البيانات المختلفة من قواعد البيانات إلى مجموعة بيانات معينة Dataset , كما تقوم بتنفيذ التعديلات التي أجريت على مجموعة من البيانات وعكسها على قواعد البيانات .

5. فئة مجموعة البيانات (Dataset) .

هي مجموعة من الأنظمة البرمجية تخزن فيها البيانات في الذاكرة وتسمح بإدارتها. أيضاً وهي عبارة عن قاعدة بيانات صغيرة موجودة في الذاكرة تمكننا من انشاء جداول وربط هذه الجداول بعلاقات وهي لاتعتمد على مصدر بيانات محدد.

6. فئة باني الأمر (CommandBuilder) .

تستخدم هذه الفئة من قبل الـ Data Adapter لاختيار وإدخال وتحديث وحذف السجلات في قاعدة البيانات , فهذه الفئة تقوم بعمل فئة الأمر ولكن على فئة من نوع مهيي البيانات .

7. فئة ربط البيانات (BindingSource) .

وهي الفئة المخصصة لربط قيمة إحدى خصائص فئة ما بإحدى خصائص أداة معينة , بحيث يتم وضع قيمة خاصية الأداة تلقائياً كلما تغيرت , والعكس ...

وباختصار تصبح الخاصيتين كأنهما خاصية واحدة . أي أن استخدام هذه الفئة في ربط محتويات إحدى خصائص الأداة ( مثل خاصية النص التابعة لمربع النص ) .

❖ هناك شيئاً آخر لابد وأن نتحدث عنه فهو مهم ولا نستطيع أن نقول عن نفسك أن بإمكانك استخدام تقنية ado.net بدون معرفة به وهي ( providers ) وهي وسيطات بين قاعدة البيانات و الاتصال ( connection ) وهي أنواع تعتمد على نوع قاعدة البيانات المستخدمة في الربط ولن أتعرق فيها كثيراً فهو للمبتدئين وسأذكر بعض أنواعها :

▪ مزود الـ SQL Server ويكتب بالشكل التالي

Provider=SqlOLEDB;

▪ مزود الأكسس

أكسس 2003

Provider=Microsoft.Jet.OLEDB.4.0;

أكسس 2007

Provider=Microsoft.ACE.OLEDB.12.0;

❖ حتى وإن لم تستوعبوا السابق بشكل جيد فإن التطبيق سيوضح لكم الأمر أكثر وسنستخدم بعض من هذه الكلاسات ( الفئات ) في مشروعنا الأول للربط وسوف ندخل إلى طرق الربط واحدة تلو الأخرى لتترسخ بالذهن فهيا بنا .



- نفتح الفيجوال ستوديو في واجهة البرنامج  
نفتح مشروع جديد بلغة السي شارب

File >> new >> project>>visual c#  
>>windows>>windows form application

ونصمم هذه الواجهة

قمت بترقيم بعض الأدوات المهمة في لأكتب لكم أسماءها لكي نفهم الكود جيداً.

1. صندوق نصي ( textbox ) وقمنا بتسميته ( txt\_no ) .

وهذا لأجل أن يقوم المستخدم برقم الصديق الذي يريد عرض بياناته في الصناديق النصية .

وأيضاً إن أراد أن يقوم بالتعديل في بيانات أحد الأصدقاء في الجدول فهنا يضع رقمه ويقوم بكتابة البيانات الجديدة في الصناديق .

2. زر ( button ) وقمنا بتسميته ( btn\_show1 ) .

هذا الزر لعرض بيانات صديق محدد يكون قد ادخل رقمه في ( txt\_no ) .

3. زر ( button ) وقمنا بتسميته ( btn\_del ) .

هذا الزر لحذف بيانات صديق محدد يكون قد ادخل اسمه في ( txt\_no ) .

4. زر ( button ) وقمنا بتسميته ( btn\_add ) .

هذا الزر يقوم بإضافة صديق جديد للجدول من البيانات المكتوبة في الصناديق الخمسة .

5. صندوق نصي ( textbox ) وقمنا بتسميته ( txt\_fri\_no ) .

رقم الصديق



6. صندوق نصي ( textbox ) وقمنا بتسميته ( txt\_fri\_name ) .  
اسم الصديق
7. صندوق نصي ( textbox ) وقمنا بتسميته ( txt\_city ) .  
المدينة
8. صندوق نصي ( textbox ) وقمنا بتسميته ( txt\_mobile ) .  
رقم الجوال
9. صندوق نصي ( textbox ) وقمنا بتسميته ( txt\_job ) .  
الوظيفة
10. زر ( button ) وقمنا بتسميته ( btn\_upd ) .  
يقوم هذا الزر بتعديل بيانات الصديق صاحب الرقم المكتوب في 1 مستخدماً  
بذلك البيانات المكتوبة في الصناديق .

الآن نبدأ

اولاً لنقوم بالربط مابين برنامج وقاعدة بيانات يجب ان نضيف فضاء الاسماء الخاص بهذا النوع من قاعدة البيانات وفضاء الاسماء هو المكاتب وتوجد في نافذة الكود في الأعلى نرى منها غالباً

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
```

وهنا نضيف المكتبة الخاصة بقاعدة البيانات التي سوف نستخدمها بالبرنامج  
نقوم بتحديد المكتبة التي من نوع قاعدة البيانات التي نشتغل عليها .  
■ في الـ SQLServer نستخدم المكتبة

```
using System.Data.SqlClient;
```

■ في الأكسس نستخدم المكتبة

```
using System.Data.OleDb;
```

الآن سنبدأ بكتابة الأكواد  
ندخل إلى نافذة الكود بالضغط على F7 وتحديدًا إلى القسم العام (public)  
نبدأ بتعريف المتغيرات وكتابة هذي الأكواد

```
public partial class Form1 : Form
{
1|     OleDbConnection con = new
OleDbConnection("Provider=Microsoft.ACE.OLEDB.12.0;Data Source=lesson7.accdb");
2|     OleDbCommand cmd;
3|     OleDbDataReader dr;
4|     DataTable dt = new DataTable();
    public Form1()
```

ما هي هذه الأكواد ؟

1. قمنا باشتقاق كائن من الكلاس OleDbConnection وأسميناه con وهذا الكلاس هو الذي يقوم بتنفيذ الاتصال بقاعدة البيانات ومن الباراميترات الخاصة به انه يستقبل المزود ومسار قاعدة البيانات فلو رأينا الكود نرى أننا أعطيناه أولاً المزود الخاص بقاعدة البيانات Provider=Microsoft.ACE.OLEDB.12.0; ومن الواضح اننا استخدمنا قاعدة بيانات اكسس 2007 .

ثم اعطيناه مسار قاعدة البيانات واسمها Data Source=lesson7.accdb وطبعاً هنا لم نحتاج لكتابة مسار قاعدة البيانات واكتفينا باسمها لأننا وضعناها في مسار البرنامج الأساسي F:\c#\2014\lesson7\lesson7\bin\Debug

2. اشتقنا كائن من فئة Command وهي الأوامر وأسميناه cmd والذي سنحمله نص الأوامر التي نحتاجها من قاعدة البيانات .

3. اشتقنا كائن من فئة DataReader وهي قارئ البيانات واسميناه dr وهو لقراءة البيانات من قاعدة البيانات .

4. اشتقنا كائن من فئة DataTable وهي فئة نستطيع أن نحملها قيم جدولية وأسميناه dt .

الآن نبدأ في باقي الأدوات

نعود الى نافذة التصميم بالضغط على SHIFT+F7

1. في الزر رقم 2 والذي قمنا بتسميته (btn\_show1). نكتب الكود التالي

```

2. try
3.     {
4.         cmd = new OleDbCommand("select * from friends where
fri_no=" +txt_no.Text + " ", con);
5.         con.Open();
6.         dr = cmd.ExecuteReader();
7.         dr.Read();
8.         txt_fri_no.Text = dr["fri_no"].ToString();
9.         txt_fri_name.Text = dr["fri_name"].ToString();
10.        txt_city.Text = dr["city"].ToString();
11.        txt_mobile.Text = dr["mobile"].ToString();
12.        txt_job.Text = dr["job"].ToString();
13.        dr.Close();
14.
15.        MessageBox.Show(" العرض تم ");
16.    }
17.    catch (OleDbException xx)
18.    {
19.        MessageBox.Show(" ما خطأ هناك " + xx.Message);
20.    }
21.    finally
22.    {
23.        con.Close();
    }

```

المهم في هذا الكود هو الأسطر من 4 الى 13 واستخدمنا دوال try و catch لتفادي الوقوع في اخطاء فقط تعالوا نشرح الكود .

4/ حملنا الكائن cmd نص الأمر و مفتاح الاتصال ونص الأمر اشتراطنا فيه

أن يكون رقم الصديق يساوي النص المدخل في ( txt\_no )

5/ فتحنا الاتصال باستخدام دالة Open(); التابعة للكائن con .

6/ أسندنا البيانات التي قام بطلبها cmd لقارئ البيانات dr

7/ جعل dr يقرأ البيانات .

8/ أدخلنا في الخاصية Text للصندوق النصوص ( txt\_fri\_no ) قيمة قمنا

بأخذها من dr وقمنا بتحديد الحقل الذي سنأخذ منه البيانات الى هذا

الصندوق بالأمر ToString(); ["fri\_no"] حيث ["fri\_no"] هي اسم الحقل

في الجدول سنقوم بعرضها في صندوق النصوص هذا ونستطيع ان نكتب

رتبة الحقل مثلاً هنا نستطيع أن نكتب 0 ولكن لتفادي الوقوع بأخطاء نكتب

اسم الحقل .

9/ وكما فعلنا مع صندوق النص الخاص برقم الصديق نفعل مع بقية الصناديق محددين كل صندوق نصي ما هو الحقل الذي سيعرضه .  
13/ نقوم بإغلاق الاتصال.

### Try & Catch & Finally

جملة **try** بالمعنى هي جرب أو حاول وهي إخبار المترجم أن يجرب أن ينفذ الكود الموجود بين الحاصرتين التابعة لها لتفادي الأخطاء ويجب بعد استخدام **try** يجب استخدام دالة **catch** والخلاصة أن **try** لها حاصرة بداية وحاصرة نهاية وما بينهما يكتب الكود الذي تريد تنفيذه ثم بعد قفل الحاصرة يجب استخدام **catch** وهي تقوم بمقام الخيار الثاني الذي ينتقل اليه المترجم إذا كان هناك خطأ بالكود الذي في داخل **try** والد **catch** ايضاً لها حاصرتين يكتب في داخلها الكود المراد تنفيذه إذا لم يتم تنفيذ جملة **try** ونستطيع أيضاً قبل فتح حاصرة الـ **catch** اشتقاق كائن من أي كلاس لنستخدمه وفي هذا المثال قمنا بأشتقاق كائن من الفئة **OleDbException** وهو بمقام الاستثناء بلغة الأس كيو إل واسمينا الكائن **xx** وجعلنا في جملة **catch** رسالة تخبرنا بالخطأ الذي وقع واستخدمنا دالة في الكائن **xx** لتعرض لنا نص الرسالة . عرفنا هنا ان جملة الـ **catch** تنفذ إذا لم تنفذ جملة الـ **try** أما الـ **finally** فهي دالة تنفذ جملتها في كلتا الحالتين سواء تنفذت جملة **try** او جمل **catch** .

الان ننتقل الى الزر الثاني

في الزر رقم 3 والذي قمنا بتسميته (btn\_del) .

نقوم بكتابة هذه الأكواد

```
cmd = new OleDbCommand("delete from friends where fri_no=" + txt_no.Text + "",
con);
```

```
con.Open();
```

```
cmd.ExecuteNonQuery();
```

```
MessageBox.Show("رقم الصديق بيانات حذف تم" + txt_no.Text);
```

في السطر الأول قمنا بأعطاء كائن الأوامر الأمر اللازم بحذف بيانات الصديق

صاحب الرقم المحدد في ال صندوق النص (txt\_no)

مع مفتاح الاتصال

ثم فتحنا الاتصال بالسطر الذي يليه

ثم قمنا بتنفيذ الأمر باستخدام الدالة **ExecuteNonQuery()**;

وهذه الدالة نستخدمها لتنفيذ الأمر مثلها مثل **ExecuteReader()**;

والفرق بينهما أن الأولى نستخدمها إذا كان هناك بيانات مرتجعة من الأمر أما

الثانية فنحن نستخدمها عندما لا يكون هناك بيانات مرتجعة مثل عمليات الإضافة

والحذف والتعديل .

وفي السطر الذي يليه قمنا بإظهار رسالة للمستخدم تفيد بأنه تم الحذف ويمكننا بعده إغلاق الاتصال ولم اكتب كود إغلاق الاتصال لأنني قمت باستخدام دوال التفادي من الأخطاء ووضعت كود الإغلاق فيها .  
ننتقل للزر الذي يليه

ننتقل للزر الذي يليه وهو الزر رقم 4 والذي قمنا بتسميته (btn\_add) ونكتب به هذا الكود

```
try
{
(1)      cmd = new OleDbCommand("insert into friends
(fri_no,fri_name,city,mobile,job) values ('" + txt_fri_no.Text + "','" +
txt_fri_name.Text + "','" + txt_city.Text + "','" + txt_mobile.Text + "','" +
txt_job.Text + "')", con);
(2)      con.Open();
(3)      cmd.ExecuteNonQuery();
(4)      MessageBox.Show(" الإضافة تمت ");
}
catch (OleDbException xx)
{
    MessageBox.Show(" خطأ " + xx.Message);
}
finally
{
    con.Close();
}
```

الآن للشرح

سأقوم بشرح الأسطر الذي قمت بترقيمها باللون الأخضر من 1 : 4 فلا داعي لأن اشرح دوال Try & Catch & Finally فأعتقد إنكم تعرفونها فقد تحدثنا عنها سابقاً

1. كما قمنا سابقاً بإعطاء الكائن cmd أمر الـ sql و مفتاح الاتصال الذي هو في مشروعنا con وهكذا فعلنا في هذا السطر قمنا باستخدام أمر الإضافة الخاص بالـ sql ولكي نأخذ القيم (values) من داخل الصناديق النصية نستخدم العلامة + الذي نستخدمها في دمج السلاسل النصية او الربط بينها ولكن يجب التركيز على التنسيقات الأحادية والمزدوجة وبعد الأمر قمنا بتسمية كائن الاتصال والذي هو con .

2. فتحنا الاتصال . con.Open();

3. قمنا بتنفيذ الأمر باستخدام الدالة cmd.ExecuteNonQuery();

4. قمنا بعرض رسالة توضيحية للمستخدم تفيد بأنه قد تمت الإضافة .

لاحظوا أننا لم نكتب كود إغلاق الاتصال؟؟ ولكننا قمنا بكتابة في دالة finally الذي ستنفذ في جميع الأحوال .

الآن ننتقل إلى الزر التالي

وهو الزر رقم 10 والذي قمنا بتسميته (btn\_upd) . والذي يقوم بحفظ التعديل ونكتب فيه هذا الكود

```
try
{
    cmd = new OleDbCommand("update friends set fri_no=" +
txt_fri_no.Text + ", fri_name='" + txt_fri_name.Text + "',city='" +
txt_city.Text + "',mobile=" + txt_mobile.Text + ",job='" + txt_job.Text + "'
where fri_no=" + txt_no.Text + "'", con);
    con.Open();
    cmd.ExecuteNonQuery();
    MessageBox.Show("التعديل تم");
}
catch (OleDbException xx)
{
    MessageBox.Show("خطأ " + xx.Message);
}
finally
{
    con.Close();
}
```

عندما نرى الكود نرى انه قد تكرر لدينا اكثر من مره وهو مشابه بدرجة كبيرة لكود الاضافة والاختلاف الوحيد هو كود الـ sql وهو يقوم بوضع القيم الجديدة من الصناديق وفق رقم الصديق المكتوب بالصندوق الأول .

ما هي البيئة المتصلة والبيئة المنفصلة؟؟؟

الدرس السابق كان في إطار البيئة المتصلة وهي التي نكون فيها على اتصال دائم بقاعدة البيانات إما البيئة المنفصلة فهي عكس ذلك أي ليست على اتصال دائم بقاعدة البيانات وإنما تأخذ نسخة من قاعدة البيانات وتشتغل بها إلى أن تتم الحاجة للاتصال

ما الفائدة من البيئتين ولماذا صممت مايكروسوفت الأدوات؟؟

إن مايكروسوفت عندما أضافت هذه التقنية واستطعنا أن نتعامل مع قواعد البيانات في الوضع المنفصل خدمة كبيرة جداً سأشرح لك لماذا وما الفائدة من جعل هناك بيئتين .

إننا نختار البيئة المتصلة عندما نكون نتعامل مع سرفر محلي وقاعدة بيانات واحدة ويستخدمها مستخدم واحد وهنا يكون الوضع المتصل أفضل للاتصال لكن إن كان هناك شبكات تتاح لأكثر من مستخدم على نفس قاعدة البيانات أي أنه في الوضع المنفصل كل التعديلات الذي تقوم بها على قاعدة البيانات تحفظ على جدول وهمي مؤقت داخل برنامجك وليس قاعدة البيانات وعند الوقت اللازم يتم حفظها إلى قاعدة البيانات

### البيئة المنفصلة

قبل ان ندخل في تطبيقات البيئة المنفصلة سنقوم بجولة بسيطة لتتعرف إلى بعض المفاهيم الذي سنحتاجها سواء كنا نتعامل في البيئة المتصلة او المنفصلة وسأذكر بعض المفاهيم التي تدخل في تقنية الـ ado.net .

## - مجموعة البيانات (Data set)

هي مجموعة من الأنظمة البرمجية تخزن فيها البيانات في الذاكرة وتسمح بإدارتها . أيضا هي عبارة عن قاعدة بيانات صغيرة موجودة في الذاكرة تمكننا من إنشاء جداول وربط هذه الجداول بعلاقات وهي لا تعتمد على مصدر بيانات محدد .

ومن بعض مكونات الـ Data set

### ✓ الفئة جدول البيانات Data Table :

عبارة عن تمثيل لجدول إما أن تقوم ببناء هيكله في Dataset أو تأخذه من مصدر بيانات ويمكن أن تحتوي Dataset على عدد غير محدد من الجداول .

### ✓ الفئة أعمدة البيانات Data Column :

عبارة عن تمثيل لعمود داخل جدول هذا العمود له نوع واسم وطول الخ ... من مواصفات الأعمدة.

### ✓ الفئة صفوف البيانات Data Row :

يمثل هذا الفئة سجل في الفئة Data Table حيث تؤمن طريقته Item الوصول الى قيم حقوله .

### ✓ الفئة قيود الأعمدة Data Constariat :

قيود توضع على الأعمدة ( تتحكم بهيئة البيانات داخل الأعمدة ) مثل (Primary , Foreign , NULL , Constariat , Unique)

### ✓ الفئة علاقات DataRelations :

نستطيع من خلاله إنشاء ( تحديد ) العلاقات بين الجداول .

### ✓ الفئة مستعرضات Data View :

إنشاء مستعرض يحتوي على جدول أو أكثر من خلال هذا الكائن .

وسنذكر هنا بعض الخصائص والكائنات الذي سنحتاجها سواء في درسنا هذا أو فيما بعد :

### ❖ الخاصية Position :

نستخدمها للتنقل بين السجلات وهي تتبع المتغير الذي تم تعريفه من نوع BindingManagerBase

### ❖ الخاصية DataBinding :

تستخدم هذه الخاصية لربط كائن مع عمود في جدول وعرض محتواه فيه .

### ❖ الخاصية BindingContext :

يستخدم لربط الكائنات المرتبطة مع الـ Dataset والجدول والتحكم بسجلات الجدول عن طريق استخدام متغير من نوع BindingManagerBase .

### ❖ الخاصية BindingManagerBase :

وهو كائن يقوم بتنسيق العلاقة بين كائنات الربط ومجموعة البيانات ويقوم بالتحكم بالحركة داخل السجلات .

### ❖ الخاصية CurrencyManager :

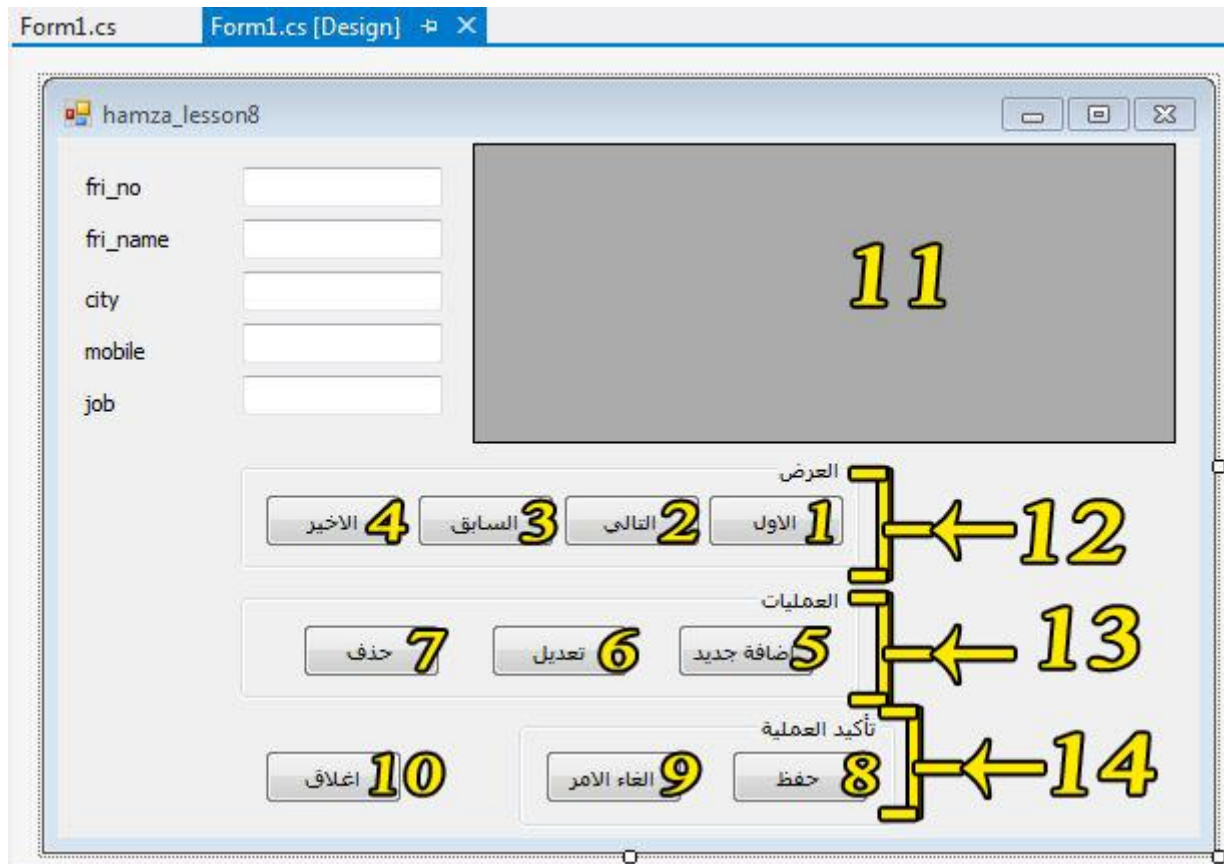
وهو كائن يقوم بنفس عمل BindingManagerBase .

### ❖ الخاصية displayMember التابعة لـ ComboBox :

تستخدم في ربط قاعدة البيانات بحقل من الجدول .



- ❖ الخاصية ValueMember التابعة لـ ComboBox :  
وهي قيمة DisplayMember التي سوف يتم إرجاعها إلى قاعدة البيانات .
  - ❖ الخاصية DataSource التابعة لـ ComboBox :  
وهي طريقة تقوم بتحديد اسم قاعدة البيانات .
  - ❖ كائن باني الأمر CommandBuilder :  
وهو كائن يستخدم من قبل الـ DataAdapter لإختيار وإدخال وتحديث وحذف السجلات في قاعدة البيانات .
  - ❖ الخاصية InsertCommand التابعة لـ DataAdapter :  
وهي خاصية بالفئة DataAdapter وتعتبر كمرجع للفئة CommandBuilder والتي يستخدمها الـ DataAdapter لإضافة البيانات بين قاعدة البيانات وفئة الـ DataSet .
  - ❖ الخاصية DeleteCommand التابعة لـ DataAdapter :  
وهي خاصية بالفئة DataAdapter وتعتبر كمرجع للفئة CommandBuilder والتي يستخدمها الـ DataAdapter لحذف البيانات بين قاعدة البيانات وفئة الـ DataSet .
  - ❖ الخاصية UpdateCommand التابعة لـ DataAdapter :  
وهي خاصية بالفئة DataAdapter وتعتبر كمرجع للفئة CommandBuilder والتي يستخدمها الـ DataAdapter لتعديل البيانات بين قاعدة البيانات وفئة الـ DataSet .
  - ❖ الخاصية SelectCommand التابعة لـ DataAdapter :  
وهي خاصية بالفئة DataAdapter وتعتبر كمرجع للفئة CommandBuilder والتي يستخدمها الـ DataAdapter لاستعلام أو اختيار البيانات بين قاعدة البيانات وفئة الـ DataSet .
  - ❖ الوظيفة GetInsertCommand التابعة لباني الأمر CommandBuilder :  
وهي عبارة عن خاصية للفئة CommandBuilder تقوم بإضافة السجلات الجديدة من الـ DataSet وإسقاطها على الـ DataAdapter عن طريقة الخاصية ( مرجع ) InsertCommand .
  - ❖ الوظيفة GetDeleteCommand التابعة لباني الأمر CommandBuilder :  
وهي عبارة عن خاصية للفئة CommandBuilder تقوم بحذف السجلات المحذوفة من الـ DataSet وإسقاطها على الـ DataAdapter عن طريقة الخاصية ( مرجع ) DeleteCommand .
  - ❖ الوظيفة GetUpdateCommand التابعة لباني الأمر CommandBuilder :  
وهي عبارة عن خاصية للفئة CommandBuilder تقوم بتعديل السجلات التي تم تعديلها من الـ DataSet وإسقاطها على الـ DataAdapter عن طريقة الخاصية ( مرجع ) UpdateCommand .
- اعتقد أن الأمور واضحة وحتى إن لم تستوعبوا السابق بشكل جيد فبالتطبيق سيزيل الالتباس والغموض وتصبح الأمور جلية أكثر هيا بنا للتطبيق نعود مرة أخرى إلى برنامج الفيجوال ستوديو ننشئ مشروع جديد ونصمم هذه الواجهة :



شرح الواجهة :

- ❖ الصناديق النصية على جهة الشمال هي كما في التطبيق السابق .
- ❖ في جهة اليمين هي أداة **dataGridView** أداة لعرض البيانات وأسميناها **dataGridView1** وهي الظاهرة معنا بالصورة بالرقم 11.

❖ مجموعة الأزرار الأولى

هي الأدوات المرقمة من (1-4) هي أزرار (**button**) وهي للقيام بعمليات التنقل بين بيانات قاعدة البيانات وأيضاً نحن وضعناها هذه الأزرار داخل أداة (**groupBox**) وأسميناها **groupBox1** وهي الظاهرة معنا في الصورة بالرقم 12.

❖ مجموعة الأزرار الثانية

هي الأدوات المرقمة من (5-7) هي أزرار (**button**) وهي للقيام بعمليات الإضافة والتعديل والحذف لبيانات قاعدة البيانات وأيضاً نحن وضعناها هذه الأزرار داخل أداة (**groupBox**) وأسميناها **groupBox2** وهي الظاهرة معنا في الصورة بالرقم 13.

❖ مجموعة الأزرار الثالثة

هي الأدوات التي أرقامها (8 و 9) هي أزرار (**button**) وهي للقيام بتأكيد عمليات الإضافة والتعديل والحذف لبيانات قاعدة البيانات وأيضاً نحن وضعناها هذه الأزرار داخل أداة (**groupBox**) وأسميناها **groupBox3** وهي الظاهرة معنا في الصورة بالرقم 14.

❖ الزر رقم (10) وهو الزر الذي يقوم بإغلاق البرنامج .

س ما أول ما نقوم به ؟

لا شك أن أول ما يجب علينا القيام به هو تعريف المكتبة التي نحتاجها وهنا نحتاج المكتبة التي تخص قاعدة البيانات أكسس والتي هي

using System.Data.OleDb;

ونبدأ تعريف المتغيرات واشتقاق الكائنات في القسم العام ونكتب التالي

```
public partial class Form1 : Form
{
    // نقوم بتعريف جملة الاتصال كما سبق ذكره قبلاً
    OleDbConnection con = new
OleDbConnection("Provider=Microsoft.ACE.OLEDB.12.0;Data
Source=lesson7.accdb");
    // تعريف كائن من فئة منظم البيانات
    OleDbDataAdapter da;
    // تعريف كائن من فئة جدول البيانات
    DataTable dt = new DataTable();
    // تعريف كائن من فئة العلاقات
    BindingManagerBase bmb;
    // تعريف كائن من فئة باني الأوامر
    OleDbCommandBuilder cmdb;
    public Form1()
    {
        InitializeComponent();
    }
}
```

سأسألكم سؤالاً لماذا وضعنا بعض الأدوات داخل الحاويات ( groupBox ) ؟؟  
للتنظيم صحيح ! أجل يمكن ولكن الغرض الذي وضعناها لأجله هو أننا نريد عند اشتغال البرنامج فإن الأزرار رقم (8 و 9) الذي يقومون بتأكيد العمليات لا نريد ان يظهرها على المستخدم الا عند اختيار المستخدم إحدى العمليات (الإضافة – التعديل – الحذف ) وتختفي بقية الأدوات وعند الضغط على أحد الزرين يعود الوضع للحالة السابقة ولهذا سنجعل الخاصية Enabled للأداة groupBox3 = False ونكتب هذه الدالة داخل البرنامج والتي سنستخدمها عند الحاجة :

```
void toolsShow()
{
    groupBox1.Enabled = !groupBox1.Enabled;
    groupBox2.Enabled = !groupBox2.Enabled;
    groupBox3.Enabled = !groupBox3.Enabled;
}
```

وهذه الدالة عند استدعائها تقوم بقلب حالة تشغيل الحاويات على ما هي عليه .

نعود إلى واجهة تصميم الواجهة بالضغط على SHIFT+F7 ونضغط على الفورم مرتين في حدث اللود نكتب الكود التالي :

```
private void Form1_Load(object sender, EventArgs e)
{
    // إعطاء كائن منظم البيانات جملة الإس كيو إل وإيضاً مفتاح الإتصال
    da = new OleDbDataAdapter("select * from friends", con);
    // استخدام الدالة فيل من دوال كائن منظم البيانات لتعبئة متغير جدول البيانات
    // أي أنه بعد تنفيذ الكود السابق يكون منظم البيانات كأنه يحمل
    // البيانات الراجعة من قاعدة البيانات ونطلب منها هنا تعبئتها للجدول ويمكننا فعل التالي بالأمر
    //dt.load(da);
    // أي أننا نجعل كائن جدول البيانات يقوم بتحميل البيانات من منظم البيانات
    da.Fill(dt);
    // وهنا نقوم بربط خاصية من خواص الأداة بحقل معين من حقول الجدول
    // باستخدام خاصية DataBindings وتستقبل بعض القيم كبارامترات
    // البارامتر الأول خاصية الأداة التي سنضيف إليها بيانات الحق وهنا استخدمنا text
    // والثاني اسم الجدول أو مصدر البيانات والذي هو عندنا dt
    // والثالث هو اسم العمود في الجدول المحدد والذي هو عندنا fri_no
    // وهذا يعني أننا قمنا بربط خاصية النص لصندوق النص هذا بالحقل المحدد في العمود رقم الصديق بالجدول
    // dt
    txtfri_no.DataBindings.Add("text", dt, "fri_no");
    // كما سبق إلا أننا هنا ربطناه بإسم الصديق
    txtfri_name.DataBindings.Add("text", dt, "fri_name");
    // وهذا ربطناه بالمدينة
    txtcity.DataBindings.Add("text", dt, "city");
    // وهذا برقم الجوال
    txtmobile.DataBindings.Add("text", dt, "mobile");
    // وهذا بالوظيفة
    txtjob.DataBindings.Add("text", dt, "job");
    // قمنا بالربط بين الـ bmb و الـ dt
    // أو نستطيع القول أننا حددنا للـ bmb الكائن الذي عليها التعامل معه
    bmb = this.BindingContext[dt];
    // قمنا بتعبئة أداة عرض البيانات ببيانات الجدول dt
    dataGridView1.DataSource = dt;
}
```

يمكنك ان تجرب تشغيل البرنامج لترى انه قد عرض لنا بيانات الجدول !!!  
و الآن نرجع إلى نافذة التصميم على الزر الأول والذي يعرض بيانات الصديق الأول  
ونكتب فيه هذا الكود

```
private void btnfrist_Click(object sender, EventArgs e)
{
    // جعلنا موقع الـ bmb = 0 أي أننا نخبره عن بيانات الصديق الذي يحتل الرتبة صفر
    bmb.Position = 0;
}
```

ونرجع إلى واجهة التصميم ونضغط على الزر الثاني والذي يعرض بيانات الصديق التالي ونكتب هذا الكود

```
private void next_Click(object sender, EventArgs e)
{
```

```
    // زدنا موقع الـ bmb بنسبة 1 أي اننا نخبره عن بيانات الصديق الذي يلي الصديق المعروض حالياً
    bmb.Position++;
}
```

ونرجع إلى واجهة التصميم ونضغط على الزر الثالث والذي يعرض بيانات الصديق السابق ونكتب الكود التالي

```
private void btnprevious_Click(object sender, EventArgs e)
{
```

```
    // نقصنا موقع الـ bmb بنسبة 1 أي اننا نخبره عن بيانات الصديق الذي يسبق الصديق المعروض حالياً
    bmb.Position--;
}
```

نرجع إلى واجهة التصميم ونضغط على الزر الرابع والذي يعرض بيانات الصديق الأخير ونكتب الكود التالي

```
private void btnlast_Click(object sender, EventArgs e)
{
```

```
    // جعلنا موقع الـ bmb = اقل من عدد صفوف الجدول بمقدار 1
    bmb.Position = dt.Columns.Count - 1;
}
```

وفي الزر رقم 5 والذي يقوم بعملية الإضافة نقوم بكتابة الكود التالي

```
private void btnAdd_Click(object sender, EventArgs e)
{
```

```
    // قمنا بإخبار المعالج اننا نريد اضافة بيانات جديدة
    bmb.AddNew();
    // استدعينا الدالة التي قمنا بكتابتها لتغيير حالة تشغيل الأزرار
    toolsShow();
}
```

وفي الزر رقم 6 والذي يقوم بعملية التعديل لا نكتب إلا كود استدعاء دالة تغيير حالة تشغيل الأزرار وفي الزر رقم 7 والذي يقوم بعملية الحذف نكتب الكود التالي :

```
private void btnDel_Click(object sender, EventArgs e)
{
```

```
    // وهذا السطر يقوم بعرض رسالة للمستخدم تقول له هل انت متأكد من عملية الحذف
    // فإن اختار نعم تحقق شرط وتم تنفيذ ما داخل الحاصرتين او بالأصح تم تنفيذ جملة If
    // وان اختار لا تم الخروج من الجملة وعدم حدوث اي تغيير
    if (MessageBox.Show("الحذف تأكيد", "الحذف عملية من متأكد انت هل",
        MessageBoxButtons.YesNo) == DialogResult.Yes)
```

```
    {
```

```
        // إعطاء أمر الحذف بالدالة Removeat والتي تقوم بالحذف حسب الموقع وتستقبل الموقع كبراميتير
        bmb.RemoveAt(bmb.Position);
        // نعطيه الأمر بإنهاء العملية الحالية
        bmb.EndCurrentEdit();
        // حددنا لباني الأوامر منظم البيانات الذي عليه التعامل معه
        cmdb = new OleDbCommandBuilder(da);
        // منظم البيانات يقوم بتحديث البيانات الذي في الـ dt
        da.Update(dt);
    }
```

```
}
```

ننتقل للزر رقم 8 والذي يقوم بحفظ أو تأكيد العملية ونكتب هذا الكود

```
private void btnSave_Click(object sender, EventArgs e)
{
    //نعطيه الأمر بإنهاء العملية الحالية
    bmb.EndCurrentEdit();
    // حددنا لباني الأوامر منظم البيانات الذي عليه التعامل معه
    cmdb = new OleDbCommandBuilder(da);
    // منظم البيانات يقوم بتحديث البيانات الذي في الـ dt
    da.Update(dt);
    // استدعينا الدالة التي قمنا بكتابتها لتغيير حالة تشغيل الأزرار
    toolsShow();
}
```

ننتقل للزر رقم 9 والذي يقوم بإلغاء العملية ونكتب الكود التالي

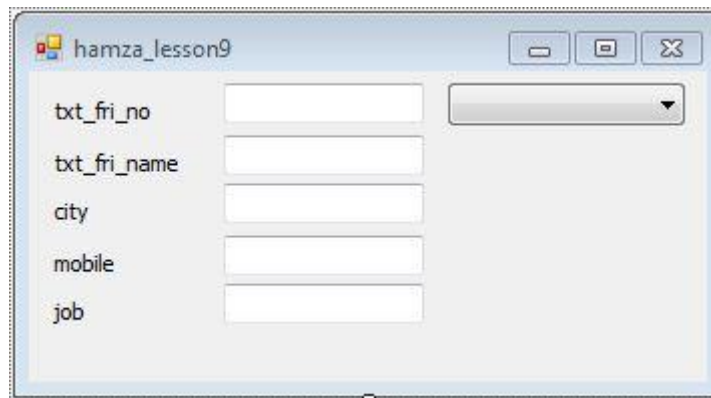
```
private void btnCancel_Click(object sender, EventArgs e)
{
    // CancelCurrentEdit(); الدالة باستخدام الدالة
    bmb.CancelCurrentEdit();
    // استدعينا الدالة التي قمنا بكتابتها لتغيير حالة تشغيل الأزرار
    toolsShow();
}
```

أما في زر الإغلاق فهو أمر إغلاق البرنامج فقط كما تعرفون

```
private void btnClose_Click(object sender, EventArgs e)
{
    this.Close();
}
```

❖ المثال السابق هو يوضح الربط ما بين بيانات قاعدة البيانات وخواص الأدوات في برنامجك في الوضع الغير متصل اعتقد أن الأمر جلياً أننا في البيئة المنفصلة نستخدم الداتا ادايتز وهو الذي يجعلنا غير متصلين بقاعدة البيانات . المهم أريد أن أضيف لكم مثلاً ضرورياً فيبدو أن العديد منكم قد تساءل لم لا نجعل طريقة اختيار المستخدم للصديق المراد عرض بياناته في قائمة الليست بوكس أو الكومبو بوكس ولهذا التساؤل بل الحاجة الملحة لتعلم ذلك هي ما جعلتني أصر أن أمر بكم على هذا الموضوع بلمحة بسيطة لتأخذوا فكرة عن هذا الموضوع وتفهموا أساسها وبعدئذ تطوروها بأنفسكم لا أطيل عليكم هيا بنا إلى المثال .

❖ نريد إن نصمم برنامج فيه كومبو بوكس يعرض لنا أسماء الأصدقاء وعند اختيارنا لأحد الأصدقاء يعرض بياناته في صناديق نصية .  
نصمم هذه الواجهة في برنامج الفيجوال ستوديو



نذهب الى الكومبو بوكس وفي الخاصية DropDownStyle نجعلها DropDownList وذلك لجعلها للقراءة فقط أو بالأصح لمنع التعديل على البيانات المعروضة عليها .  
ثم ندخل إلى نافذة الكود وبعد كتابة فضاء الأسماء الخاص بقاعدة البيانات ننزل إلى تعريف المتغيرات واشتقاق الكائنات في الجزء العام ونكتب الكود التالي :

```
// نقوم بتعريف جملة الاتصال كما سبق ذكره قبلاً
OleDbConnection con = new
OleDbConnection("Provider=Microsoft.ACE.OLEDB.12.0;Data Source=lesson7.accdb");
// اشتقاق كائن فئة منظم أو جالب البيانات
OleDbDataAdapter da;
// تعريف كائن من فئة جدول البيانات ليستقبل البيانات الراجعة
// وهذا الذي سنستخدمه في تعبئة الكومبو بوكس
DataTable dt = new DataTable();
// اشتقاق كائن آخر من فئة جدول البيانات ليستقبل البيانات الراجعة
// وهذا الذي سنستخدمه في تعبئة الصناديق النصية
// لأن هذا الجدول سيحمل بيانات سطر واحد أما السابق فهو يحمل بيانات عمود
DataTable dt2 = new DataTable();
public Form1()
{
    InitializeComponent();
}
```



أظن أن الشرح واضحاً الآن نعود إلى نافذة التصميم ونضغط على الفورم مرتين لننتقل إلى حدث اللود أي حدث تحميل الفورم ونكتب الكود التالي :

```
private void Form1_Load(object sender, EventArgs e)
{
    // إعطاء كائن منظم البيانات أو جالبها جملة الإس كيو إل وإيضاً مفتاح الإتصال
    da = new OleDbDataAdapter("select * from friends", con);
    /*
    استخدام الدالة فيل من دوال كائن منظم البيانات لتعبئة متغير جدول البيانات
    أي أنه بعد تنفيذ الكود السابق يكون منظم البيانات كأنه يحمل
    البيانات الراجعة من قاعدة البيانات ونطلب منها هنا تعبئتها للجدول ويمكننا فعل التالي بالأمر
    dt.load(da);
    أي أننا نجعل كائن جدول البيانات يقوم بتحميل البيانات من جالب البيانات
    */
    da.Fill(dt);
    // ثم نحدد مصدر بيانات الكومبو بوكس أنها من الجدول
    comboBox1.DataSource = dt;
    // نحدد الحقل الذي نريد إظهاره في الكومبو بوكس
    // وهنا حددنا الحقل رقم الصديق باستخدام خاصية DisplayMember للكومبو بوكس
    comboBox1.DisplayMember = "fri_name";
    // نكتب اسم الحقل صاحب المفتاح الرئيسي في جدول الأصدقاء في الخاصية ValueMember للكومبو بوكس
    //
    comboBox1.ValueMember = "fri_no";
}
```

الآن نحن قد حملنا الكومبو بوكس ببيانات العمود اسم الصديق لجميع الأصدقاء ! جرب التنفيذ وسترى.  
نعود إلى نافذة التصميم وننقر على الكومبو بوكس نقرتين لندخل إلى حدث SelectedIndexChanged لها ونكتب الكود التالي :

```
private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
{
    try
    {
        dt2.Clear();
        da = new OleDbDataAdapter("select * from friends where fri_name='"
+comboBox1.Text + "'", con);
        // لاشك اننا نعرف ان الإستعلام السابق يرجع لنا سطر من البيانات
        // وهنا قمنا بتعبئة الـ dt2 ببيانات هذا السطر
        da.Fill(dt2);
        /*
        الآن نأتي لأظهار بيانات السطر الذي في الجدول في الصناديق النصية
        نحدد الجدول ثم نستخدم دالة ROWS وهي دالة تحدد الصف وتستقبل رتبة السطر
        ورتبة السطر عندنا هو الصفر لأن الإستعلام السابق يجلب سطر واحد فقط
        وتستقبل أيضاً رتبة الحقل ورتبة الحقل في رقم الصديق هي 1
        ولكن لتفادي الأخطاء نكتب اسم الحقل أفضل من رتبته
        وعلى نفس الطريقة نكمل باقي الصناديق النصية
        */
        txt_fri_no.Text = dt2.Rows[0]["fri_no"].ToString();
        txt_fri_name.Text = dt2.Rows[0]["fri_name"].ToString();
        txt_city.Text = dt2.Rows[0]["city"].ToString();
        txt_mobile.Text = dt2.Rows[0]["mobile"].ToString();
        txt_job.Text = dt2.Rows[0]["job"].ToString();
    }
    catch
    {
        return;
    }
}
```



ونستطيع أيضاً إضافة بيانات للكومبو بوكس او الليست بوكس بعدة طرق منها بالبيئة المتصل باستخدام الداتا ريدر بهذه الطريقة

```
private void button1_Click_1(object sender, EventArgs e)
{
    try
    {
        cmd = new OleDbCommand("select * from friends", con);
        con.Open();
        dr = cmd.ExecuteReader();
        comboBox1.DisplayMember = "fri_name";
        comboBox1.ValueMember = "fri_no";
        while (dr.Read())
        {
            comboBox1.Items.Add(dr["fri_name"].ToString());
        }
        dr.Close();

        MessageBox.Show("الاضافة تم");
    }
    catch (OleDbException xx)
    {
        MessageBox.Show("ما خطأ هناك" + xx.Message);
    }
    finally
    {
        con.Close();
    }
}
```

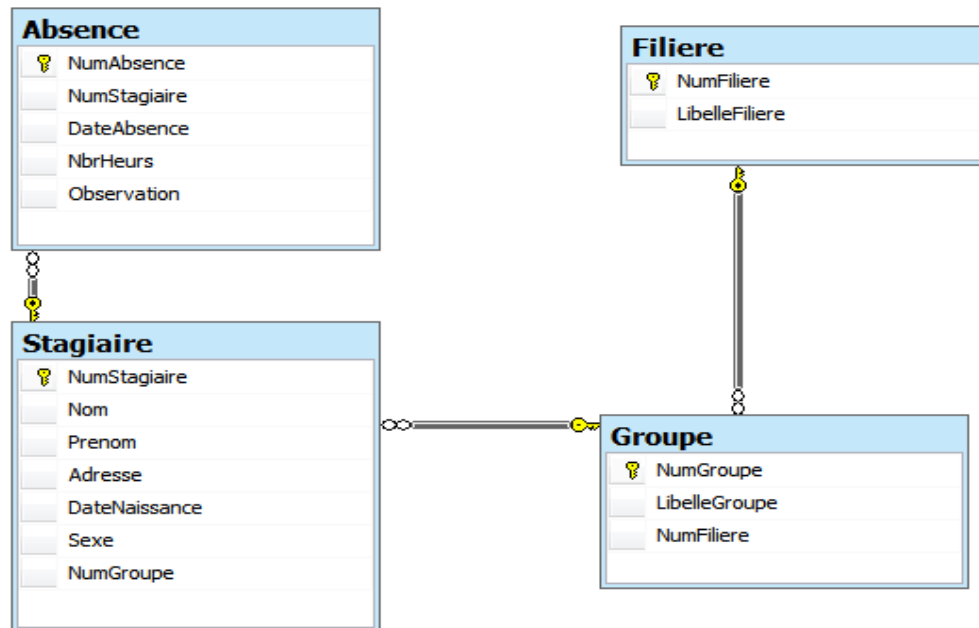
- ❖ يجب أن أنبهكم أننا في الدروس السابقة كلها كنا نضع قاعدة البيانات داخل مجلد البرنامج وان لم تكن هناك يجب علينا كتابة موقع قاعدة البيانات ولوضعها في مسار البرنامج كما كنا نفعل طيلة الدروس السابقة نفتح مجلد البرنامج (bin) ومن داخله ندخل مجلد (Debug) ونضعها هنا لكي تكون في مسار البرنامج .
- ❖ وإن تريدون أن لا تكتبوها انتم تستطيعون من واجهة الفيجوال الضغط على تبويب (Data) واختيار (Add New Data Source) ثم الضغط على زر (Next) ثم نضغط ( New Connection ) ثم من زر (Browser) ثم نحدد مكان قاعدة البيانات التي نريد استخدامها ونضغط (Open) لديك في النافذة زر مكتوب عليه ( Test Connection ) لاختبار نجاح الاتصال ثم الضغط على (OK) فتفتح لنا الواجهة السابقة التي تحمل اسم ( Data Source Configuration Wizard ) نرى في الأسف أداة تفرع عليها علامة + ومكتوب عليها (Connection String) نضغط على علامة + فيظهر لنا أسفلها كود الاتصال نستطيع من هنا أن نقوم بنسخة ولصقه في مكان تحديد نص الاتصال .

أعتذر إذا قصرت في بعض الأشياء ولكني قد قلت لكم في البداية هذا الكتاب هو مقدمة فقط ولكنه يحوي أهم الأشياء في هذه التقنية وتقبلوا خالص تحياتي لأي استفسار يرجى مراسلتي على احد الإيميلات وانتظرونا في الكتب القادمة .

X

حمزه علي البعسي  
برمجة حاسوب - اليمن

بالنسبة لقاعدة البيانات فهذه صورتها :



كما تلاحظ هناك أربعة جداول:

- **Stagiaire** : وهو جدول الطلاب و يضم الحقول التالية (رقم الطالب، الاسم، النسب، العنوان، تاريخ الازدياد، الجنس) أما رقم المجموعة فهو حقل أجنبي للربط بين الطالب و المجموعة التي ينتمي إليها.
- **Groupe** : جدول المجموعات و يضم الحقول التالية ( رقم المجموعة، اسمها) أما رقم التخصص NumFiliere فهو حقل أجنبي للربط بين المجموعة و التخصص



• **Filiere** : جدول التخصصات و يضم الحقول التالية (رقم التخصص، اسمه).

• **Absence** : وهو جدول الغياب و يضم الحقول الآتية (رقم الغياب و سنجدله تلقائي

increment، و تاريخ الغياب، وعدد الساعات المتغيبه، و الملاحظات).

كخلاصة، قاعدة البيانات هذه ما هي إلا نموذج مبسط سنسعى من خلاله شرح المفاهيم المرومة.

و هذا هو كود إنشاء قاعدة البيانات :

```
create database Stagiaire
GO
use Stagiaire;

--table Filière
create table Filiere (NumFiliere int, LibelleFiliere varchar(100),
Primary key (NumFiliere))

--table Groupe

create table Groupe (NumGroupe int, LibelleGroupe varchar(50), NumFiliere int
, Primary key (NumGroupe), Foreign key (NumFiliere) references Filiere)

--table stagiaire

create table Stagiaire (NumStagiaire int, Nom varchar(5), Prenom varchar(50),
Adresse varchar(250), DateNaissance datetime, Sexe varchar(50), NumGroupe
int,
primary key (NumStagiaire), Foreign key (NumGroupe) references Groupe)

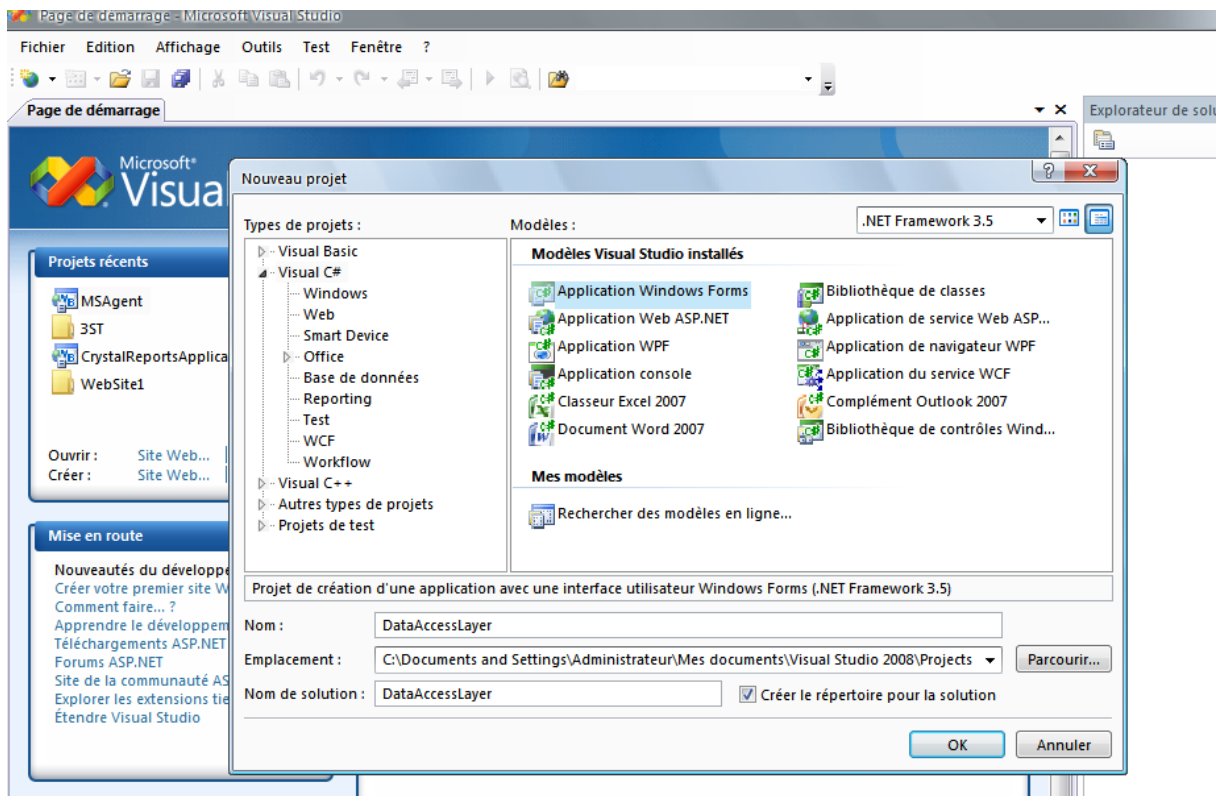
--table Absence

create table Absence (NumAbsence int, NumStagiaire int, DateAbsence Datetime,
NbrHeurs int, Observation nvarchar(250), Primary key (NumAbsence),
Foreign key (NumStagiaire) references Stagiaire)
```



الآن بعد أن أنشأت قاعدة البيانات إذهب إلى الفيجوال استوديو و قم بإنشاء مشروع جديد و ليكن

اسمه DataAccessLayer :



بعد ذلك سنقوم بإنشاء الفئة الرئيسية DataAccessLayer ولكن قبل ذلك تعال معي لتعرف عليها



الداتا أكسيس لاير هي قبل كل شيء فئة Class و من خلال اسمها نستنبط أنها تعني (طبقة الدخول إلى البيانات). بمعنى أنها القنطرة الرابطة بين البرنامج و بين قاعدة البيانات.

إذا لاحظت معي فقد ذكرنا هنا ثلاثة عناصر مهمة وهي :

- قاعدة البيانات

- الداتا أكسيس لاير

- البرنامج (أقصد بالبرنامج النوافذ و الفئات)

و هذا ما يسمى التطبيق متعدد الطبقات، أو ما يصطلح عليه ب MVC (Model,View,Controller)

الموديل هنا هو قاعدة البيانات، الفيو هو البرنامج، و الكونترولر هو الداتا أكسيس لاير.

و الغاية من هذه البنية هي تنظيم البرمجة و تقسيمها إلى أجزاء ليسهل التعامل مع كل جزء على حدى، و حتى نركز الشفرة الرئيسية في فئة واحدة تكون الجسر الفاصل بين البيانات و بين البرنامج، هذه الفئة التي ذكرنا بأنها الداتا أكسيس لاير.

من هنا نستنتج بأن الداتا أكسيس لاير ستضم دوال الربط بقاعدة البيانات و قراءة المعطيات و إرسالها.

سأفترض أنك على اطلاع بالإجراءات المخزنة stored procedure في sql server . إن كان الأمر كذلك فلنتابع إذن.

سننشئ فئة جديدة باسم الداتا أكسيس لاير .



و سنقوم باستيراد مجالات الأسماء التالية : System.Data.SqlClient, System.Data

وسنقوم بالاعلان عن متغير من نوع SqlConnection

بعد ذلك سننشئ مشيدا للفتة Constructor ومن خلاله سنحدد نص الكونيكشن.

أي سيكون الكود هكذا :

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Data.SqlClient;
using System.Data;
namespace DataAccessLayer
{
    class DataAccessLayer
    {
        static SqlConnection SqlCon;

        public DataAccessLayer()
        {
            SqlCon = new SqlConnection("Server=TECHNI-AMECO-PC\\SQLEXPRESS;Database=Stagiaire;integrated security=true");
        }
    }
}
```

واحذر أن تقوم بنسخ الكود فعليك أن تغير نص الكونيكشن حسب نظامك.

بعد ذلك سننشئ دالتي ربط الإتصال و إغلاقه :



```
public void Connect()
{
    if (SqlCon.State != ConnectionState.Open)
        SqlCon.Open();
}

public void Disconnect()
{
    if (SqlCon.State != ConnectionState.Closed)
        SqlCon.Close();
}
```

و في الأخير سننشئ دالة لقراءة البيانات و دالة لإرسال البيانات :

```
public DataTable SelectData(String Stored, SqlParameter[]
Param)
{
    SqlCommand SqlCmd = new SqlCommand(Stored, SqlCon);
    SqlCmd.CommandType = CommandType.StoredProcedure;
    for (int i = 0; i < Param.Length ; i++)
    {
        SqlCmd.Parameters.Add(Param[i]);
    }
    SqlDataAdapter SqlDa = new SqlDataAdapter(SqlCmd);
    DataTable Dt = new DataTable();
    SqlDa.Fill(Dt);
    return Dt;
}
```

هذه الدالة تعيد لنا جدولاً DataTable لنتمكن من استغلاله في إظهار البيانات و اللعب بها كما نريد.





تأخذ الدالة متغيرين داخليين أولهما Stored وهو اسم الإجراء المخزن، و Param و هي البارامترات التي تنتظرها Stored Procedure.

و سنفهم هذا بالتفصيل فيما يلي.

ثم أعلننا عن متغير من نوع SqlCommand ليرسل البارامترات إلى الإجراء المخزن وذلك بتمريرها كلها عن طريق التكرار For.

بعد عملية إرسال البارامترات، ننشئ كائنا من SqlDataAdapter ليقوم بأخذ نتائج SqlCommand ووضعها في Datatable التي سنحصل نحن فيما بعد على البيانات من خلالها.

و هذه دالة إرسال البيانات :

```
public void ExecuteCommand(String Stored, SqlParameter[] Param)
{
    SqlCommand SqlCmd = new SqlCommand(Stored, SqlCon);
    SqlCmd.CommandType = CommandType.StoredProcedure;
    for (int i = 0; i < Param.Length; i++)
    {
        SqlCmd.Parameters.Add(Param[i]);
    }
    SqlCmd.ExecuteNonQuery();
}
```

وهي مثل الدالة السابقة إلا أنها لا تقوم بقراءة البيانات و إنما بإرسالها.

أتمنى أن تكون قد تعرفت على الداتا أكسيس لاير ، إن كان العكس فلا تتردد في قراءة الفقرة جيدا فالأمر سهل جدا ويستلزم منك فقط شيئا من التركيز ، وهذا هو الكود كاملا:



```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Data.SqlClient;
using System.Data;
namespace DataAccessLayer
{
    class DataAccessLayer
    {
        static SqlConnection SqlCon;

        public DataAccessLayer()
        {
            SqlCon = new SqlConnection("Server=TECHNI-AMECO-PC\\SQLEXPRESS;Database=Stagiaire;integrated security=true");
        }

        public void Connect()
        {
            if (SqlCon.State != ConnectionState.Open)
                SqlCon.Open();
        }

        public void Disconnect()
        {
            if (SqlCon.State != ConnectionState.Closed)
                SqlCon.Close();
        }

        public DataTable SelectData(String Stored, SqlParameter[] Param)
        {
            SqlCommand SqlCmd = new SqlCommand(Stored, SqlCon);
            SqlCmd.CommandType = CommandType.StoredProcedure;
            for (int i = 0; i < Param.Length ; i++)
            {
                SqlCmd.Parameters.Add(Param[i]);
            }
            SqlDataAdapter SqlDa = new SqlDataAdapter(SqlCmd);
            DataTable Dt = new DataTable();
            SqlDa.Fill(Dt);
            return Dt;
        }
    }
}
```



```
public void ExecuteCommand(String Stored, SqlParameter[] Param)
{
    SqlCommand SqlCmd = new SqlCommand(Stored, SqlCon);
    SqlCmd.CommandType = CommandType.StoredProcedure;
    for (int i = 0; i < Param.Length; i++)
    {
        SqlCmd.Parameters.Add(Param[i]);
    }
    SqlCmd.ExecuteNonQuery();
}
}
```

و الآن حان الوقت لنستغل الداتا أكسيس لاير، سنقوم بإنشاء فئة Absence تحتوي على دالتين : الأولى مشاهدة غياب الطالب من خلال رقمه، و الثانية إضافة غياب جديد للطالب:

هذه هي الدالة الأولى:

```
public DataTable SelectAbsence(int NumStagiaire)
{
    DataAccessLayer Dal = new DataAccessLayer();
    DataTable Dt = new DataTable();
    SqlParameter[] Param=new SqlParameter[0];
    try
    {
        Param[0] = new SqlParameter("NumStg", SqlDbType.Int);
        Param[0].Value = NumStagiaire;
        Dal.Connect();
        Dt=Dal.SelectData("SP_SelectAbsence", Param);
    }
    catch
    {
        Dal.Disconnect();
    }
    return Dt;
}
```



بالنسبة لل SP\_SelectAbsence فهي إجراء مخزن Stored Procedure يقوم بجلب بيانات غياب الطالب من خلال رقمه و هذا هو كود الإجراء المخزن :

```
create proc SP_SelectAbsence
@NumStg int
as
select DateAbsence,NbrHeurs,Observation
from Absence where NumStagiaire=@NumStg
```

أعتقد أن الإجراء يشرح نفسه بنفسه، بحيث يتلقى رقم الطالب كبارامتر و يبحث عنه.

الدالة الثانية في الفئة Absence وهي دالة إضافة غياب جديد و لكن قبل ذلك سننشئ الإجراء المخزن الذي يقوم بعملية الإضافة:

```
create proc SP_AddAbsence
@NumStg int,@DateAbsence datetime,@NbrHeur int,@Observation nvarchar(250)
as
insert into absence (NumStagiaire,DateAbsence,NbrHeurs,Observation)
values (@NumStg,@DateAbsence,@NbrHeur,@Observation)
```

يعني قمنا بإضافة البارامترات التي يتلقاها الإجراء المخزن إلى الجدول Absence.

والآن هذه هي دالة الإضافة و التي تستعمل الداتا أكسيس لاير كوسيلة للوصول إلى الإجراء المخزن للكتابة في الجدول Absence.



```
public void AddAbsence(int NumStagiaire, DateTime DateAbsence, int
NbrHeurs, String Observation)
{
    DataAccessLayer Dal = new DataAccessLayer();
    SqlParameter[] Param = new SqlParameter[4];
    try
    {
        Param[0] = new SqlParameter("NumStg", SqlDbType.Int);
        Param[0].Value = NumStagiaire;
        Param[1] = new SqlParameter("DateAbsence",
SqlDbType.DateTime);
        Param[1].Value = DateAbsence;
        Param[2] = new SqlParameter("NbrHeur", SqlDbType.Int);
        Param[2].Value = NbrHeurs;
        Param[3] = new SqlParameter("Observation",
SqlDbType.NVarChar);
        Param[3].Value = Observation;
        Dal.Connect();
        Dal.ExecuteCommand("SP_AddAbsence", Param);
    }
    catch
    {
        Dal.Disconnect();
    }
}
```

كما ترى قمنا بإرسال البارامترات إلى الإجراء المخزن عن طريق الوسيط الداتا أكسيس لاير. نرسل كل بارامتر بعدما نحدد اسمه ونوعه و يلزم أن يكونا مطابقين لمثيليهما في الإجراء المخزن، و أدنى اختلاف قد يسبب خطأ.

ونرصد هنا شفرة الفئة Absence.



```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Data.SqlClient;
using System.Data;
namespace DataAccessLayer
{
    class Absence
    {
        public DataTable SelectAbsence(int NumStagiaire)
        {
            DataAccessLayer Dal = new DataAccessLayer();
            DataTable Dt = new DataTable();
            SqlParameter[] Param=new SqlParameter[1];
            try
            {
                Param[0] = new SqlParameter("NumStg", SqlDbType.Int);
                Param[0].Value = NumStagiaire;
                Dal.Connect();
                Dt=Dal.SelectData("SP_SelectAbsence", Param);
            }
            catch
            {
                Dal.Disconnect();
            }
            return Dt;
        }
    }
}
```



```
public void AddAbsence(int NumStagiaire, DateTime DateAbsence, int
NbrHeurs, String Observation)
{
    DataAccessLayer Dal = new DataAccessLayer();
    SqlParameter[] Param = new SqlParameter[4];
    try
    {
        Param[0] = new SqlParameter("NumStg", SqlDbType.Int);
        Param[0].Value = NumStagiaire;
        Param[1] = new SqlParameter("DateAbsence",
SqlDbType.DateTime);
        Param[1].Value = DateAbsence;
        Param[2] = new SqlParameter("NbrHeur", SqlDbType.Int);
        Param[2].Value = NbrHeurs;
        Param[3] = new SqlParameter("Observation",
SqlDbType.NVarChar);
        Param[3].Value = Observation;
        Dal.Connect();
        Dal.ExecuteCommand("SP_AddAbsence", Param);
    }
    catch
    {
        Dal.Disconnect();
    }
}
}
```

والآن حان موسم جني الثمار، كل ما قمنا بإنشائه من كود سنستعمله الآن ببساطة منقطعة النظير، حيث في أسطر قليلة سنقوم بالكثير من المهام وهذه نقطة من أهم إيجابيات الداتا أكسيس لاير.



ليكن الفورم التالي هو فورم الغياب:

سنقوم بإنشاء شفرة إضافة غياب و ذلك بالضغط مرتين على الزر Ajouter.

```
Absence Ab = new Absence();  
Ab.AddAbsence(int.Parse(TxtID.Text),  
DateAbsence.Value,int.Parse(TxtNbr.Text), TxtObserv.Text);
```

لا تستغرب فهذا هو كود إضافة غياب جديد .





أما كود البحث فهو كما يلي:

```
Absence Ab = new Absence();  
DataTable Dt = new DataTable();  
Dt=Ab.SelectAbsence(int.Parse(TxtSearch.Text));  
this.DGV.DataSource = Dt;
```

أرأيت جمال الداتا أكسيس لاير !

ويمكن أن نعلن عن المتغير Ab مرة واحدة وهذا هو الكود الكامل :

```
using System;  
using System.Collections.Generic;  
using System.ComponentModel;  
using System.Data;  
using System.Drawing;  
using System.Linq;  
using System.Text;  
using System.Windows.Forms;  
namespace DataAccessLayer  
{  
    public partial class Form1 : Form  
    {  
        Absence Ab = new Absence();  
        public Form1()  
        {  
            InitializeComponent();  
        }  
  
        private void button2_Click(object sender, EventArgs e)  
        {  
            Ab.AddAbsence(int.Parse(TxtID.Text), DateAbsence.Value,  
int.Parse(TxtNbr.Text), TxtObserv.Text);  
        }  
    }  
}
```



```
private void button1_Click(object sender, EventArgs e)
{
    DataTable Dt = new DataTable();
    Dt=Ab.SelectAbsence(int.Parse(TxtSearch.Text));
    this.DGV.DataSource = Dt;
}
}
```

كنت أود أن أتعامل مع الداتا أكسيس لاير من خلال فئات جديدة لها علاقة بالتخصصات و المجموعات ولكنني الآن مقتنع تماما بأن نفس المبدأ سينطبق عليها لذا ارتأيت الإكتفاء بهذا فحسب.

لا تتشائم من مظهر الداتا أكسيس لاير فبشيء من التطبيق ستجد نفسك قد استسغتها كليا بل و أصبحت قادرا على أن تبدع فيها و بها.